On Quasi-reducibility for c.e. sets
Part II. Relationships to other reducibilities

Sapir Ben-Shahar Rodney G. Downey Mariya Soskova*

August 5, 2026

Abstract

We explore the relationship between the following reducibilities on computably enumerable sets of natural numbers: many-one, quasi, strong quasi, weak truth table and Turing. Let s- and w- denote a pair of a stronger and weaker reducibility among the ones listed. We study for which pairs there is a w-degree that is contiguous with respect to s-reducibility, and show that there are incomplete high m-contiguous Q-degrees. We investigate under what circumstances a w-degree can be s-topped. We investigate under what circumstances an s-minimal pair can be extended to a w-minimal pair. We use this to show that the c.e. Q-degrees have infinitely many one-types.

1 Introduction

1.1 Background

A central goal of computability theory is determining the relative complexity of problems. This field originated in the 1930's with Church, Turing, and Kleene, who proved that certain mathematical problems are undecidable. To prove this, they typically encoded a known undecidable problem—such as the halting problem—into the problem under review, a technique used in Turing's proof of the undecidability of the Entscheidungsproblem [29]. This

*The third author is partially supported by NSF Grant No. DMS-2451099 and Grant SFI-MPS-TSM-00013602

approach established an implicit hierarchy ($P_1 \leq P_2$), meaning P_2 is at least as difficult as P_1 because a solution to P_2 provides a method for solving P_1. Turing formalized this concept in 1939 with Turing reducibility ($\leq_T$) [30].

Following Post's landmark 1944 paper [25], researchers introduced various other reducibilities based on different "oracle access" mechanisms. Well-studied examples include $\leq_m$ (many-one), $\leq_{tt}$ (truth table), $\leq_{wtt}$ (weak truth table). Recall that for sets of natural numbers A and B[1]:

- $A \leq_m B$ if and only if there is a computable function f such that for all x, $x \in A$ if and only if $f(x) \in B$;
- $A \leq_{tt} B$ if and only if there is a Turing procedure Φ total for all oracles such that $x \in A$ if and only if $\Phi^B(x) = 1$, and
- $A \leq_{wtt} B$ if and only if there is a computable function g and a Turing procedure Ψ such that Ψ^B is total, $x \in A$ if and only if $\Psi^B(x) = 1$, and additionally for all y the use of the computation is bounded by $g(y)$.

The definition here of $\leq_{tt}$ is not the original one, which asks for a computable function h such that $h(x)$ is a boolean condition and $x \in A$ if and only if B satisfies this condition. The original definition allows us to have variations such as disjunctive truth table reducibility (*dtt*-) and conjunctive truth table reducibility (*ctt*-). For both we compute a canonical finite set $D_{k(x)}$ and $A \leq_{dtt} B$ if and only if $\exists z \in D_{k(x)}(z \in B)$ and $A \leq_{ctt} B$ if and only if $D_{k(x)} \subseteq B$. Evidently $\leq_m$ implies $\leq_{tt}$ implies $\leq_{wtt}$ implies $\leq_T$, which is why these more refined reducibilities are called *strong reducibilities*. (More information about strong reducibilities can be found in Odifreddi [21].)

Strong reducibilities are useful for many reasons. On the one hand, they offer granular insight into localized applications where Turing reductions are too coarse: for example, $\leq_{tt}$ preserves measures in algorithmic randomness, and Overbeek [24] proved that every c.e. m-degree is the degree of a word problem for a finitely presented semigroup. Second, they illuminate the broader structure of $\leq_T$. A prime example is the study of "contiguous" degrees (Turing degrees comprised of a single *wtt*-degree), which allows structural properties to be transferred from *wtt*-degrees to Turing degrees [1, 14]. *Both* of these reasons are valid for the reducibilities we study in the present paper.

[1]For simplicity in this paper, we will always assume that $A, B \notin \{\emptyset, \mathbb{N}\}$.

In the present paper we study $\leq_Q$ and its strong refinement $\leq_{sQ}$:

Definition 1 (Tennenbaum)**.** We say that A is *quasi-reducible* to B and write $A \leq_Q B$ if and only if there is a computable function f such that for all x,

$$x \in A \text{ if and only if } W_{f(x)} \subseteq B.$$

Soloviev [28] observed that if A and B are c.e. then $\leq_Q$ refines $\leq_T$. It can be seen as a generalization of $\leq_{ctt}$ since we are replacing canonical finite sets with indices of c.e. sets.

Quasi-reducibility established its proper place in the study of computability theory through its relevance to Post's programme. Post's programme asked for a "thinness" property in the lattice of c.e. supersets of a c.e. set which ensured Turing incompleteness. Post's motivation stemmed from a key observation regarding the undecidability proofs of his era: all known codings [25] fundamentally worked by hard-coding the halting problem into the target problem, typically via m-reductions. Post noted that because infinitely many machines do not halt, any set $A \subseteq \mathbb{N}$ to which the halting problem m-reduces must possess infinitely many c.e. subsets in its complement, $\overline{A}$. This intuition was later formalized by Myhill [20], who demonstrated that all m-complete c.e. sets are creative. Thus a possible path to solve Post's Problem, i.e. to find a c.e. degree $\mathbf{a}$ that is neither computable nor complete, is to look for a c.e. set with a "very thin" complement. It is now known that there is no single definable property in the lattice of c.e. supersets that ensures that a c.e. non-computable set is Turing incomplete [7] (See Odifreddi [21] for the full story). Nevertheless, Post's programme can be realized and this is where Q-reducibility comes into play. Soloviev [28] and Gill and Morris [15] showed that no maximal c.e. set M can be Q-complete. So if we could construct such a set M where for c.e. W, $W \leq_T M$ if and only if $W \leq_Q M$ then we'd have a solution to Post's problem. In this case, we would say that the Turing degree of M is *Q-topped* by M. If M were semi-recursive then this would be true. This almost works, however, a generalization of the notion of maximality is needed. Consider the lattice of c.e. sets $\mathcal{E}$ replacing $=^*$ by a c.e. equivalence relation η. Degtev [8] constructed an "η-maximal semirecursive set" and Marchenkov [19] showed that this could not be Q-complete. That is, for suitably constructed η, an η-maximal semirecursive set had the property that for c.e. W, $W \leq_T M$ if and only if $W \leq_Q M$, and additionally it had to be Q-incomplete and hence T-incomplete.

Another well-known application of Q-reducibility is in combinatorial group theory. If H is a finitely generated group then let $W(H)$ denote its *word problem*, i.e. the set of words over the given generators that are equal to the identity. Belegradek [5] established that if H_1 and H_2 are finitely generated nontrivial c.e. groups, then H_1 is embeddable in any existentially closed group that H_2 embeds in if and only if $W(H_1) \leq_Q W(H_2)$. Consequently, H_1 is embeddable into every existentially closed group if and only if H_1 has a solvable word problem.

Omanadze introduced a natural strong variant of $\leq_Q$:

Definition 2 (Omanadze [23])**.** We say that $A \leq_{sQ} B$ (*strongly* Q-reducible) if there is a Q-reduction $x \in A$ if and only if $W_{f(x)} \subseteq B$, and a computable function g such that

$$\max\{z \mid z \in W_{f(x)}\} \leq g(x).$$

Thus $\leq_{sQ}$ is the analog of weak truth table reducibility for Q-reducibility. Indeed, for c.e. sets A and B we have the following implications:

$$A \leq_m B \implies A \leq_{sQ} B \begin{array}{c} \nearrow \\ \searrow \end{array} \begin{array}{c} A \leq_Q B \\ A \leq_{wtt} B \end{array} \begin{array}{c} \searrow \\ \nearrow \end{array} A \leq_T B$$

This paper is the second in a series that investigates the properties of $\leq_Q$, $\leq_{sQ}$ and the degree structures that arise from them. In the first paper [6] we focused on algebraic properties of the structure of the Q-degrees and the sQ-degrees. For example, we showed that these structures are not distributive and that the lattice N_5 embeds into both. In the present paper we investigate various relationships between $\leq_Q$, $\leq_{sQ}$ and other well studied reducibilities: namely $\leq_T$, $\leq_{wtt}$ and $\leq_m$.

1.2 The Present Paper

The classical notion of a *contiguous degree* is a c.e. Turing degree that consists of a single c.e. *wtt*-degree. Contiguous degrees were first constructed by Ladner and Sasso [18], and were used to understand the structure of the c.e. Turing degrees. For example, Ambos-Spies and Soare [3] used them

to show that the c.e. Turing degrees have infinitely many one-types. All contiguous c.e. degrees are low_2, and they are downward dense.

It is not hard to see that we cannot have a Turing degree that consists of a single m-degree, however, we can weaken the definition slightly and consider *m-topped* degree. The c.e. degree of a set A is m-topped if for all c.e. $X \leq_T A$, $X \leq_m A$. Naturally enough, $\emptyset'$ has this property, but Downey and Jockusch [12] showed that there are incomplete c.e. m-topped degrees, they are all low_2 and not low, and Downey and Shore [10] proved that if X is a low_2 c.e. set, then there is an incomplete A with $X \leq_T A$ and $\deg_T(A)$ m-topped.

We can talk about contiguity and being topped whenever we have two reducibilities $s-$ and $w-$ between c.e. sets such that $A \leq_s B$ implies $A \leq_w B$. We give the following definition:

Definition 3. (i) We say that a c.e. w-degree is *s-contiguous* if for all c.e. members X, Y of the w-degree, $X \equiv_s Y$.

(ii) We say that a c.e. w-degree $\mathbf{a}$ is *s-topped* if there is a c.e. set $A \in \mathbf{a}$ such that for all c.e. sets $X \leq_w A$ implies $X \leq_s A$.

Batyrshin [4] established the remarkable result that there are m-contiguous c.e. Q-degrees[2]. This is really quite surprising. Jockusch [17] proved that every c.e. Turing degree contains infinitely many c.e. m-degrees. Further, every non-computable c.e. wtt-degree contains infinitely many c.e. tt-degrees, so m-contiguous wtt-degrees also do not exist. On the other hand, Degtev [8] constructed a c.e. m-contiguous tt-degree, and Downey [11] studied the distribution of these degrees within the c.e. Turing degrees.

In §2, we prove that every c.e. Turing degree contains an sQ-contiguous c.e. Q-degree. We also show that sQ-contiguous c.e. wtt-degrees cannot exist, nor can Q-contiguous c.e. Turing degrees, so the existence or non-existence of contiguity is now known for all combinations of m, sQ, Q, wtt and T. Examining m-contiguous c.e. Q-degrees slightly closer, we show that there are incomplete high sets whose Q-degree is m-contiguous.

On the other hand, in §3 we show that no c.e. semi-low set can have its Q-degree m-topped and hence m-topped and m-contiguous Q-degrees must be non-low. We show that there are non-low Turing degrees containing no m-topped Q- or sQ-degrees. We also show that if $\mathbf{a}$ is a c.e. wtt-degree and

[2]Batyrshin used the term m-singular instead of m-contiguous

$A \in \mathbf{a}$ is c.e. semirecursive, then $W \leq_{wtt} A$ implies $W \leq_{sQ} A$ for any c.e. W. That is, semirecursive sets are the sQ-tops of their wtt-degree.

In §4 we consider minimal pairs and sets that are half of a minimal pair. We show that for c.e. sets $A \equiv_{wtt} B$, if the infimum of A and B in the Q- or sQ-degrees exists, it will have the same wtt-degree as A and B. That is, we cannot have a Q- or sQ-minimal pair within the same wtt-degree. This is in contrast to the result of Downey, LaForte and Nies [9] that there are Q-minimal pairs within the same Turing degree. We then show that, nevertheless if $\mathbf{a}$ is half of a minimal pair in the c.e. sQ-degrees, it is also one in the T-degrees.

Finally, in §5 we prove that if $\mathbf{b}$ is a c.e. Q-degree which bounds a minimal pair then so does its T-degree. This allows us to show that the c.e. Q-degrees have infinitely many one-types.

1.3 Preliminaries

We recall that $A \leq_Q B$ if and only if there is a computable function f such that for all x, $x \in A$ if and only if $W_{f(x)} \subseteq B$. We also recall that $A \leq_{sQ} B$ if and only if additionally there is a computable function h such that $\max\{z : z \in W_{f(x)}\} < h(x)$.

From Downey et al. we know that for c.e. sets we can assume that $W_{f(x)}$ is finite, and at every stage s, either $x \in A_s$ or

- there is a unique $z \in W_{f(x)} \setminus B[s]$.
- if $W_{f(x)}[s] \neq W_{f(x)}[s+1]$, then $W_{f(x)}[s] \subseteq B[s+1]$. (So, in particular, the z at stage $s+1$ cannot be any earlier z.)

We can define a total computable function $m(x,s)$ to be Q-*like* if $m(x,s) = z$ and hence satisfies the restrictions above. In other words, $m(x,s) \in B[s]$ should only be possible if $x \in A$, and if $m(x,s) \neq m(x,s+1)$ then $m(x,s) \in B[s+1]$. That is, for c.e. sets A and B we have $A \leq_Q B$ if and only if there is a Q-like m such that $\lim_s m(x,s) = m(x)$ exists and $m(x) \in B$ if and only if $x \in A$.

These also hold for sQ, and furthermore for all s, $m(x,s) < h(x)$.

2 Contiguous Degrees

In this section we examine the notion of contiguity for pairs of a strong and a weak reducibility among our list. First we consider Q-degrees within the same Turing degree. In 1987 [22] Omanadze proved that every non-computable c.e. Turing degree contains a c.e. set whose Q-degree contains neither simple nor nowhere simple sets. He also showed that any c.e. set that is Q-reducible to a nowhere simple set is itself nowhere simple. As a result, every c.e. set contained in the Q-degree of a nowhere simple set is nowhere simple. That is, a c.e. Q-degree consists either entirely of nowhere simple sets, or does not contain any nowhere simple sets. On the other hand, Shore proved in 1978 [26] that every c.e. Turing degree contains a nowhere simple set. Since this set cannot have the same Q-degree as Omanadze's set, we have the following simple observation.

Observation 4. *Every c.e. Turing degree contains at least two distinct Q-degrees. That is, Q-contiguous Turing degrees do not exist.*

We now show that c.e. *wtt*-degrees cannot be sQ-contiguous.

Proposition 5. *For every c.e. non-computable A, there is a c.e. $B \equiv_{wtt} A$ such that $A \not\leq_{sQ} B$.*

Proof. Given a non-computable c.e. set A, construct a c.e. set $B \equiv_{wtt} A$ such that $A \not\leq_{sQ} B$. To do this, we meet the requirements

$$R_e : m_e \text{ is not a } Q\text{-like function}$$

We do not need to use the bounded nature of sQ-reductions here, so in fact we are constructing B such that $A \not\leq_Q B$. In order to maintain that $B \equiv_{wtt} A$, whenever an element n_s enters A at stage s, we need to enumerate an element into B below some known bound, say below $2(n_s + 1)$. We will assume that at most one element n_s enters A at stage s.

To meet an R_e requirement, we monitor $m_e(n_s, s)$ and if it converges, we want to restrain this number $m_e(n_s, s)$ from ever entering B. Say that an R_e requirement *requires attention* at stage s if $m_e(n_s, s)\downarrow > 2e$, $m_e(n_s, s) \notin B_s$ and R_e does not already have an element that it is restraining.

Construction. At stage s, if R_e requires attention, restrain $m_e(n_s, s)$ out of B. If n_s exists at stage s (so an element did enter A at this stage), enumerate the largest unrestrained element below $2(n_s + 1)$ into B.

Verification. First notice that each R_e can only restrain an element that is greater than $2e$, and requirements restrain only one element. Hence below $2e$ there are at most e-many restraints, so we always have unrestrained elements to enumerate. To check that R_e is met, assume towards a contradiction that m_e is total and is a Q-like function. If at some stage R_e restrains an element, then this element witnesses that m_e is not a Q-like function and we are done. If R_e never restrains an element, then we claim A is computable. For R_e to never restrain an element, it must be that whenever an element $n = n_s$ is enumerated into A at stage s, either $m_e(n, s) \in B_s$, or $m_e(n, s) \leq 2e$. To compute $A(n)$, go to a stage s such that B has stopped changing below $2e$. If $m_e(n, s) \in B_s$ we know $n \in A$. If $m_e(n, s) \leq 2e$ then $A(n) = A_s(n)$ since B won't change below $2e$ and so A cannot change on n. If $m_e(n, s) > 2e$ then again $A(n) = A_s(n)$, because since we are not in one of the previous cases, if n is not already in A and were to enter A at some stage $t > s$, we must have $m_e(n, t)\uparrow$ (otherwise R_e would require attention and restrain $m_e(n, t)$), but this cannot happen since m_e is total. Thus R_e is met. □

On the other hand, we *do* have sQ-contiguous c.e. Q-degrees in every non-computable c.e. Turing degree.

Theorem 6. *sQ-contiguous c.e. Q-degrees exist in every non-computable c.e. Turing degree.*

Proof. Let X be a given non-computable c.e. set with $f(\omega) = X$ a 1-1 computable enumeration of X. We build $A = \cup_s A_s$ with $A \equiv_T X$ to meet the requirements:

$$N_e : [A \leq_Q V_e \text{ via } m_e(\cdot, \cdot) \text{ and } V_e \leq_Q A \text{ via } n_e(\cdot, \cdot)] \to V_e \equiv_{sQ} A.$$

Recall that m_e witnesses that $A \leq_Q V_e$, if it is a Q-like function: $m_e(x, s) = z \notin V_e[s]$ unless $x \in A$, and if $m_e(x, s + 1) \neq m_e(x, s)$ then $m_e(x, s) \in V_e[s + 1]$. So m_e only changes its marker if the previous one goes in. Furthermore, $\lim_s m_e(x, s) = m_e(x)$ exists and $m_e(x) \in V_e$ if and only if $x \in A$. Similarly n_e is Q-like and gives the reduction in the other direction, $n_e(x) \in V_e$ if and only if $x \in A$. So if x enters A at stage s, $m_e(x, s)$ must enter V_e, and m_e stops changing. While $x \notin A$, $m_e(x, s) \notin V_e$.

The *length of agreement* is

$$\ell(e,s) = \max\{z : \forall x \leq z((x \in A_s \leftrightarrow m_e(x,s) \in V_{e,s}) \wedge \\ \forall p \leq m_e(x,s)(p \in V_{e,s} \leftrightarrow n_e(p,s) \in A_s))\}$$

We regard s as *expansionary* if $\ell(e,s) > n_e(m_e(m\ell(e,s)))$ where $m\ell$ is the maximum length of agreement seen so far, up to the previous stage.

Coding markers and boxes. To make $A \equiv_T X$ at each stage s we will have coding markers $a_{i,s} \in \overline{A}_s$ and they will have associated disjoint *boxes* $B(i,s) = \{a_{i,s}, \ldots, a_{i+d(i,s),s}\}$ where $d(i,s)$ is a fast growing function that we will specify below. We will always have that if i is not coded (i.e. $i \notin X_s$) then $B_{i,s} \subseteq \overline{A}_s$;

If $i \in X_{s+1} \setminus X_s$ (i.e. $f(s+1) = i$), then we must put a unique element from $B_{i,s}$ into $A_{s+1} \setminus A_s$. We would then declare that i to be coded. Precisely which element of $B_{i,s}$ is chosen is part of the construction below.

Once an element from $B_{i,s}$ is chosen, in order of not-yet-coded $j > i$ we will move $a_{j,s}$ and hence the box $B_{j,s}$ to be bigger than any element ever seen hitherto, and re-set $d(j, s+1) = \sum_{e \leq j}(\delta(\ell(e, s+1)) + |B_e| + |B_{e+1}| + \cdots + |B_j|)$, where $\delta(\ell(e,s)) = \max\{m_e(x,s) : x \leq m\ell(e,s)\}$. We call this the *Box Kick*.

Note that we *only* move a box $B_{j,s}$ (making it much bigger) if i enters $X_{s+1} \setminus X_s$ for some $i < j$. This means that $\lim_s B_{j,s} = B_j$ exists. Moreover, since we code from box $B_{i,s}$ if i enters X, it follows that $A \equiv_T X$.

Construction. To meet N_e, we will build Q-like functions $\hat{m} = \hat{m}_e$ and $\hat{n} = \hat{n}_e$ such that if $\ell(e,s) \to \infty$ then $\hat{m}$ and $\hat{n}$ give the sQ-reductions $A \leq_{sQ} V_e$ and $V_e \leq_{sQ} A$, respectively. At stage s, for $e < s$ we will monitor $\ell(e,s)$.

Fix some $v \in V_e$ and $u \notin V_e$.

For s an e-expansionary stage, we scan all $z \leq \ell(e,s)$ such that z is initially in block B_i for $i \geq e$. We restrict to $i \geq e$ so that only finitely many requirements can restrain elements in each block.

If $z \in A_s$, then set $\hat{m}(z,s) = v$.

If $z \notin A_s$ and $\hat{m}(z,t)$ has not yet been defined for any $t < s$, then set $\hat{m}(z,s) = m_e(z,s)$. We refer to this as the *initial value*. If $n_e(m_e(z,s),s)$ is in block B_j where $j > i$ then we restrain $n_e(m_e(z,s),s)$ out of A. By making $d(i,s)$ grow fast enough, we will ensure that the block B_j has suf-

ficiently many elements so that the restrained elements do not exhaust B_j. Note that if ever $m_e(z,s)$ enters V_e then we will have met this requirement since $n_e(m_e(z,s),s)$ is restrained, resulting in $n_e(\cdot,\cdot)$ not actually being a Q-like function. Otherwise, the restraint keeps $m_e(z,s)$ out of V_e at further expansionary stages.

If $\hat{m}(z,t)$ was defined at stage t and $m_e(z,s) = m_e(z,t)$ we keep $\hat{m}$ the same, so $\hat{m}(z,s) = \hat{m}(z,t)$. The length of agreement ensures that the computation is correct.

Finally, consider if $m_e(z,s) \neq m_e(z,t)$ where t is the stage in which $\hat{m}(z,t)$ was first defined. That is, the initial value $m_e(z,t)$ enters V_e on or before stage s. Since we have an expansionary stage, this is only possible if the corresponding marker $n_e(m_e(z,t),t) < \ell(e,t)$ has entered A at some stage r where $t < r < s$. Let us consider what lead to enumerating $n_e(m_e(z,t),t)$ in A. Suppose $n_e(m_e(z,t),t)$ is in the j-th block B_j (so j must have entered X, hence we enumerated into A something from block B_j). By our assumptions we have $j \leq i$ as otherwise $n_e(m_e(z,t),t)$ was restrained out of A. If $j < i$ then the box B_i was kicked at stage r and hence z will never enter A. We set $\hat{m}(z,s)$ to be $u \notin V_e$, fixed in advance. If $j = i$, then $n_e(m_e(z,t),t)$ is the unique element in block B_i that is in A, and so z will never enter A. Set $\hat{m}(z,s) = u$.

Similarly, when s is an expansionary stage we scan $w \leq \delta(\ell(e,s))$, where $\delta(\ell(e,s)) = \max\{m_e(x,s) : x \leq m\ell(e,s)\}$. Let a be some number that we commit to be in A, and $\hat{a}$ be a number that we keep out of A.

If $w \in V_{e,s}$ then set $\hat{n}(w,s) = a$.

If $w \notin V_{e,s}$ and $\hat{n}(w,t)$ is not yet defined for any $t < s$, let $\hat{n}(w,s) = n_e(w,s)$ (the *initial value*).

If the initial value was defined at stage t and $n_e(w,s) \neq n_e(w,t)$ then since the length of agreement has returned (due to s being expansionary), we have $n_e(w,t) \in A_s$ and $n_e(w,s)$ is a new number not in A_s. If $n_e(w,s) > n_e(w,t)$ then $n_e(w,s)$ is in a larger block and this strategy will put a restraint on it to ensure that it never enters A, and so w can never enter V_e and we can hence redefine $\hat{n}(w,s) = \hat{a}$. We will show that if $n_e(w,t)$'s block is bigger than e then this restraint is honored. This means that after the stage at which $X \upharpoonright e$ has stabilized, all restraints of this strategy will be honored. If $n_e(w,s) < n_e(w,t)$ then we set $\hat{n}(w,s) = n_e(w,s)$.

Note that we can consider the initial value for $\hat{m}$ and $\hat{n}$ as the bounds

for these reduction, so $\hat{m}$ and $\hat{n}$ are sQ-reductions.

Now at stage $s+1$, when we see i enter $X_{s+1} \setminus X_s$, we need to decide which element of $B_{i,s}$ we need to put into $A_{s+1} \setminus A_s$. We claim the following:

Claim 7. *There is a way to computably define the function $d(i,s)$ so that if i enters X at stage s then there is an element in $B_{i,s}$ that is not restrained by any strategy with index smaller than i.*

Assuming Claim 7, choose the least such z, and put $z \in A_{s+1} \setminus A_s$.

Verification. All we need to do is prove that Claim 7 holds. Since no one can restrain numbers in B_0 we define the initial value of $d(0,0) = 1$. Suppose we have defined $d(i,0)$. Then who can restrain numbers in block B_{i+1}: a strategy R_e where $e \leq i$ can restrain one number for every member of B_e, $B_{e+1} \dots B_i$ during the initial definition of $\hat{m}$. Then it can restrain a second number for every $w \leq \delta(\ell(e,s))$ during the final definition of $\hat{n}$ at stage s. Thus if we set $d(i+1,0)$ to be $\sum_{e \leq i}(\delta(\ell(e,0)) + |B_e| + |B_{e+1}| + \dots + |B_i|)$ we will satisfy the initial requirement. What can happen is that very many w's have the same initial value, but then later they each change to a different $n(w,s)$ when the initial value goes in, and then the requirement will want to restrain all of these $n(w,s)$'s. However when the initial value goes in, we will do a box kick, and redefine the box widths $d(j,s+1)$ to be large enough to account for all the possible $n(w,s)$'s that the requirement wants to restrain, that is, for all the w's that we have considered so far. Since a number is only restrained if it is *larger* than the initial value, we can't overfill earlier boxes by having many w's with the same initial value changing to different $n(w,s)$'s that are smaller than the initial value, since these will not be restrained by the strategy.

To kick the i-th box at stage s, we redefine $a_{i,s+1}$ (and correspondingly all $a_{j,s+1}$ for $j > i$) to be a large number never seen before in the construction, and redefine $d(j,s+1)$ for $j \geq i$ to be $d(j,s+1) = \sum_{e \leq j}(\delta(\ell(e,s+1)) + |B_e| + |B_{e+1}| + \dots + |B_j|)$. Notice that we do not change the size of the previous box, B_{i-1}, so now we have a gap in the boxes: numbers between $a_{i-1,s} + d(i-1,s)$ and $a_{i,s}$ that are not in any box. This is because some numbers larger than the box B_i may have been enumerated into A at an earlier stage, and we want the box B_{i-1} to only contain a single unique element of A. The numbers $a_{i-1,s} + d(i-1,s), \dots, a_{i,s}$ which are not already in A_s will never be enumerated into A. For each $j \geq i$ that is already in X_s, we pick a new unique element from block B_j (all of which will now be

unrestrained) to put into A. $\square$

Looking at m-contiguous Q-degrees, whose existence was established by Batyrshin [4], we show that these can be high and incomplete.

Theorem 8. *There exists an incomplete high m-contiguous Q-degree.*

Proof. We build c.e. sets A and E, and a Turing functional Γ to satisfy the following list of requirements:

S_e: If $W_e \leq_Q A$ via m_e and $A \leq_Q W_e$ via n_e, then there are computable functions g and f such that $W_e \leq_m A$ via g and $A \leq_m W_e$ via f.

H_x: The s-limit of $\Gamma^A(x,s)$ is defined and equals 1 if φ_x is total, 0 otherwise.

I_e: The set E is not Turing reducible to A via Φ_e.

Basic Strategies. Consider a contiguity requirement S_e in isolation. The strategy to satisfy it is to monitor the length of agreement between W_e and A via m_e and n_e. At each expansionary stage s we define g and f on more numbers below the length of agreement $\ell(e,s)$. The length of agreement is the largest initial segment $\ell(e,s)$ such that for all $x < \ell(e,s)$ we have that $x \in W_{e,s}$ if and only if $m_e(x,s) \in A_s$, and $x \in A_s$ if and only if $n_e(x,s) \in W_{e,s}$. When we see the length of agreement go above x, we define $g_e(x)$ to be large and choose a fresh marker $z(x)$ for x. The idea is that when the length of agreement goes above $z(x)$, say at stage t, either we can use $z(x)$ to diagonalize S_e, or we will define $f_e(x)$ to be $n_e(z(x),t)$ and promise that $x \in A$ if and only if $z(x) \in A$. We can diagonalize if $m_e(n_e(z(x),s),s) \neq z(x)$, because $n_e(z(x),s)$ cannot enter W_e unless we put $m_e(n_e(x,s),s)$ into A, but on the other hand if we put $z(x)$ into A then $n_e(z(x),s)$ must enter W_e in order for n_e to be a valid Q-reduction. So if $m_e(n_e(z(x),s),s) \neq z(x)$, we can put $z(x)$ into A and keep $m_e(n_e(z(x),s),s)$ out of A and thus satisfy this requirement because now m_e and n_e fail to witness that $A \equiv_Q W_e$ (diagonalization gets a bit more complicated when we consider multiple requirements, as discussed below). If at any point we have the opportunity to diagonalize, we will do so, provided this does not ruin any higher priority requirement.

Suppose that y needs to be added to A at stage s. There might be several x with $m_e(x,s) = y$, and these x may themselves be $x = f(z)$ for some z. For each of these, if y enters A then it is possible that x enters W_e

or that $m_e(x, s+1)$ is redefined to a different number not in A. If the latter happens we don't need to do anything. However, if x enters W_e, we need to correct g by enumerating $g(x)$ in A, and if $x = f(z)$ then we also need to correct f by enumerating z into A. But now there may be other numbers w with $m_e(w, s) = g(x)$ or $m_e(w, s) = z$. And so one number entering A sets off a chain reaction. Further, for the length of agreement to recover, if we put y into A then $n_e(y, s)$ will have to enter W_e, potentially requiring us to enumerate an additional number or two into A to correct f and/or g, which could also set off chain reactions. At any point we have only defined finitely many numbers so we can wait until all has settled again and we have functions that are working as expected. We will not add additional g or f definitions until the effects of a chain reaction have been settled.

Consider the highness requirement H_x. At stage s we define $\Gamma^A(x, s) = 0$ with a very, very large use. If we see that $\varphi_x(0)$ is defined, we must change $\Gamma^A(x, t_0)$ to 1 by enumerating into A some number that is less than or equal to the use $\gamma^A(x, t_0)$. The number t_0 in this case can be defined as 0, but later will depend on higher priority requirements. We use this opportunity to kick up the use on all numbers $t > t_0$. If later we see that $\varphi_x(1)$ is defined then we enumerate some number in A that is less than $\gamma^A(x, t_1)$, where $t_1 = t_0 + 1$, we also kick up the use of all numbers $t > t_1$ and so on.

Finally, consider the requirement I_e. It tries to ensure that $E \neq \Phi_e^A$, so it picks a number n, and waits until $\Phi_e^A(n)$ is defined and equals 0. When this happens it enumerates n in E and preserves A on the initial segment up to the use $\varphi_e(n)$.

We will make use of a tree of strategies and order the requirements linearly as usual. Nodes on the tree will have two outcomes $\infty <_L w$. (Technically, I_e-strategies have two finitary outcomes, but for ease of notation we will denote their left outcome by ∞).

Interactions between requirements. Here is a problematic interaction that prevents this construction from working for the Turing degrees. Consider an S_e-strategy σ and an I-strategy $\tau \succeq \sigma^\frown\infty$, where ∞ is the outcome that signifies that σ's length of agreement goes off to infinity. If some H-strategy (say in-between) enumerates a number x in A at stage s, then this allows a number y targeted for W_e to reset its marker $m_\sigma(y, s+1) = m_e(y, s+1)$ to some larger number x_1. Then x_1 might be enumerated in A by the H-strategy again at some later $t > s$, moving $m_\sigma(y, t+1)$ even further. The strategy τ may see a computation that

includes $g(y)$ as part of its use, but arbitrarily late σ may decide to enumerate $g(y)$ into A thereby injuring τ. This can happen infinitely many times with a larger and larger set of y's. Additionally, if y itself is $y = f(w)$ for some w which is included in the use of τ's computation, then if y enters W_e when $m_\sigma(y, s)$ is enumerated into A, then σ will enumerate w into A also, thereby injuring τ. The main idea here is to make the Γ-use so big that if $m_\sigma(y, s)$ is allowed to retarget, then its new value will never enter A. So when an H-strategy ρ enumerates a number x in A, it looks up at all S_e-strategies σ that are currently active and checks how many y currently have $m_\sigma(y, s) = x$, and how many w have $m_\sigma(f(w), s)$ defined. It ensures that the use of $\Gamma^A(z, t)$ is larger than this number for any new definition of Γ. Thus next time we need to redefine $\Gamma^A(z, t)$ we can enumerate a number in A that is not any new value of $m_\sigma(y, t)$ or any current $m_\sigma(f(w), s)$.

Another interaction occurs between different S_e-requirements. Say we have S_e and S_i, with $e < i$, so S_e has higher priority than S_i. Suppose we are looking at a version of S_i below the infinite outcome of S_e. Now S_i no longer has to have $m_i(n_i(z(x), s), s) = z(x)$ in order to stop us from diagonalizing; it could instead have $m_i(n_i(z(x), s), s) = g_e(n_e(z(x), s))$, or $m_i(n_i(z(x), s), s) = g_e(y)$ for some y (potentially not in the range of n_e) for which $m_e(y, s) = z(x)$. More generally, $m_i(n_i(z(x), s), s)$ could be any element that enters A as part of the chain reaction that is started by $z(x)$ entering A. For instance, S_e might have a chain: w_1 with $m_e(w_1, s) = z(x)$ and we have defined $g_e(w_1) = p_1$, and then w_2 with $m_e(w_2, s) = p_1$ and we have defined $g_e(w_2) = p_2$, and so on, ending with w_k and p_k for some k. This chain also extends further because of n_e, for example there could be $v_1 = n_e(p_1, s)$ and we have defined $g_e(v_1) = p_{k+1}$, and there could be $v_2 = n_e(p_{k+1}, s)$ and w_{k+1} with $m_e(w_{k+1}, s) = p_{k+1}$ and so on. Then it could be that $m_i(n_i(z(x), s), s) = p_j$ for some p_j in this chain. Now say we put $z(x)$ into A. If w_1 enters by the next e-expansionary stage after we put $z(x)$ in, then S_e will enumerate p_1 into A. Then $n_e(p_1, s)$ has to enter W_e for there to be another e-expansionary stage, and at the next e-expansionary stage w_2 may have entered W_e. If it does then S_e puts p_2 into A, and so on. The chain reaction may eventually cause S_e to put $p_j = m_i(n_i(z(x), s), s)$ into A, in which case we are not able to diagonalize S_i.

To see why we need to use $z(x)$ and define $f_i(x) = n_i(z(x), s)$ instead of just defining $f_i(x) = n_i(x, s)$ directly, consider what happens if we did this when $x = g_e(u)$ for some u not in the range of n_e (so we cannot force u to enter W_e by enumerating an element in A). So x cannot enter A

unless u enters W_e, else we ruin the definition of g_e. Now it could be that $m_i(n_i(x,s),s) = p_j$ enters A as part of a chain reaction caused by some other element entering A, allowing $n_i(x,s) = f_i(x)$ to enter W_i, and n_i can then retarget. Because of S_e restraining x out of A, this ruins our definition of f_i without giving us any opportunity to diagonalize. Using $z(x)$ for our definition of $f_i(x)$ fixes this, because $z(x)$ is chosen fresh and any values of $g_e(u)$ are chosen to be large when g_e is defined, so we will never have $z(x) = g_e(u)$, and thus are always free to put $z(x)$ into A as this will not cause any problems for S_e (except for when $z(x)$ is restrained by some higher priority I-requirement, but these restrain only a finite amount of A and so this can only happen finitely many times). Now, when the length of agreement goes above $z(x)$ we consider $m_i(n_i(z(x),s),s)$. If the chain reactions set off by $z(x)$ entering do not implicate $m_i(n_i(z(x),s),s)$ (that is, no higher priority S_e-requirement will put $m_i(n_i(z(x),s),s)$ into A as part of a chain reaction set off by $z(x)$ entering), then we diagonalize S_i by putting $z(x)$ into A and keeping $m_i(n_i(z(x),s),s)$ out. At this point we do not define $f_i(x)$ because if we have succeeded in diagonalizing we have no obligation to define f_i. However, it is possible that some other element entering A at some later stage causes a chain reaction that forces $m_i(n_i(z(x),s),s)$ to enter A, allowing m_i and n_i to recover. In this case we would define a new proxy $z(x)$ for x that is fresh and repeat.

On the other hand, if the chain reaction associated to $z(x)$ *does* implicate $m_i(n_i(z(x),s),s)$ so we cannot diagonalize, then we define $f_i(x) = n_i(z(x),s)$ and promise $z(x) \in A$ if and only if $x \in A$. In this case, the only way that we are forced to put $m_i(n_i(z(x),s),s)$ into A is if we put x and $z(x)$ in A (possibly as part of some other chain reaction), and if $x = g_e(u)$ for some u this will only happen if u first entered W_e. So if we cannot diagonalize, we can guarantee that $f_i(x) = n_i(z(x),s)$ enters W_i if and only if x (and hence $z(x)$) enters A.

Construction. At stage s we build a finite path δ_s of length s on the tree. $\delta_s \upharpoonright 0$ is always the root. If $\delta_s \upharpoonright n$ is defined and $n < s$ then we let the strategy associated with this node act as described next and select an outcome $o \in \{\infty, w\}$. Then $\delta_s(n) = o$ and all strategies to the left of $\delta_s \upharpoonright n+1$ are initialized. If $n = s$ then we end stage s and move on to stage $s+1$.

Suppose that $\sigma = \delta_s \upharpoonright n$ is an S_e-strategy. We assign disjoint sets R_σ of even numbers to each S-strategy σ and choose g-markers from that set. We define the length of agreement $\ell(\sigma, s)$ to be the greatest l such that for all

$x < l$ we have $x \in W_e[s]$ if and only if $m_e(x,s) \in A[s]$, and $x \in A[s]$ if and only if $n_e(x,s) \in W_e[s]$. The stage s is considered an expansionary stage if $\ell(\sigma, s)$ is larger than all numbers that were mentioned by σ at the previous σ-true stage. If the stage s is not expansionary then σ has outcome w with no further action.

If the stage s is expansionary and σ sees an opportunity to diagonalize then σ diagonalizes and ends with outcome w. Otherwise, σ considers all numbers $x < \ell(\sigma, s)$ in turn. If x has already been previously considered at some stage $t < s$ then σ defined $g(x) \in R_\sigma$ to be large and has picked some large $z(x)$ marker for x. If $z(x)$ is itself below the length of agreement for the first time and we cannot diagonalize, then define $f(x) = n_\sigma(z(x), s)$. If $x \in W_{e,s}$ but $g(x)$ is not in A_s then we enumerate $g(x)$ in A and end with outcome w, but first we initialize all strategies extending this outcome. It will follow from the construction that no other strategy may enumerate $g(x)$ in A. If $x \notin A_s$ and $f(x) \in W_{e,s}$ then we enumerate x in A and end with outcome w, initializing all strategies extending this outcome. Otherwise we move on to $x + 1$. (Note that once a chain reaction has been started, we do not define more g and f values until everything has settled back into place.)

If x has not yet been considered by σ then we define $g(x) \in R_\sigma$ to be a number that is larger than any considered in the construction so far, and define $z(x)$ to be fresh. If all numbers have been scanned without resulting in the outcome w then we end with outcome ∞.

Suppose that $\delta_s \upharpoonright n = \rho$ is an H_x-strategy. If this is the first time that we visit this strategy we set $t_\rho = t_0$ to be larger than any number seen so far and one such that $\Gamma(x, t_0)$ has not yet been defined. We define $\Gamma(x, t_0) = 0$ with large use (defined below). Otherwise, if $\varphi_x[s]$ is defined on the same initial segment as at the previous stage when we visited this strategy (this is not the case if this is the first visit after initialization) then for all t such that $\Gamma^A(x,t)[s]$ is undefined set $\Gamma^A(x,t)[s] = 1$ if $t < t_\rho$ and $\Gamma^A(x,t)[s] = 0$ if $t \geq t_\rho$. In each case the use will only consider odd numbers and will have the same length as before: $\gamma^A(x,t)[s] = \gamma^A(x,t)[r]$, where r is the last stage when $\Gamma^A(x,t)[r]$ was defined. Let the outcome be w. Otherwise, if $\varphi_x[s]$ is defined on a larger initial segment than when we last visited ρ then check whether $\Gamma^A(x, t_\rho)[s]$ is defined. If so then enumerate in A the largest odd number $u < \gamma^A(x, t_\rho)$ that is not the m_σ value of any number x so that (σ, x) is protected by this strategy. Let L be the number of tuples (σ, z) or $(\sigma, f(z))$ such that the strategy σ has defined a g-value or an f-value for z respectively (note σ has arbitrary priority relative to ρ) or will do

so at this stage. Define $\Gamma^A(x, t_\rho)[s] = 1$ with large use and then redefine $\Gamma^A(x, t)[s] = 0$ for all $t > t_\rho$ with large new uses. We set $t_\rho = t_\rho + 1$. Here a *large use* is bigger than the previous use by at least $2(t_\rho + L + 1)$. The strategy ρ also promises to protect all such tuples from now on.

Finally if $\delta_s \upharpoonright s$ is an I_e-strategy τ that has not been visited so far then τ picks a new witness $n_\tau \notin E[s]$. We check whether $\Phi_e^A[s](n_\tau) = 0$ with a *believable* use. Here a use is believable if it does not contain any numbers that may be enumerated later in A by a higher priority strategy. When τ sees a computation, it can predict this: a number a may enter A if a higher priority H-strategy puts it in, as lower priority H-strategies visited at further stages are in initial state currently and will select their parameter t_0 and its first use large enough to protect this computation. The numbers that higher priority H-strategies enumerate in A are larger and larger, so τ will only believe a computation involving numbers that are smaller than the ones that higher priority H-strategies with the infinite outcome are working with. The other option is that a higher priority S_e strategy σ with $\sigma^\frown\infty \preceq \tau$ enumerates the number a in A. This means that $a = g(x)$ for some x and x enters W_e, or that $f(a)$ enters W_e. By construction, assuming that τ will be visited again this means that first of all $g(x)$ (or $f(a)$ in the second case) is already defined (otherwise σ will define it larger than the current computation) and second of all the current value of $m_e(x, s)$ (or $m_e(f(a), s)$) has just entered A to allow for this. This is because from now on all H-strategies that enumerate numbers in A at further stages will protect (σ, x) and $(\sigma, f(a))$. Numbers of this kind are also increasing in size. Indeed, we will argue that if $g(x)$ does not enter A by the next σ-expansionary stage, it never will.

If there is such a computation then we enumerate n_τ in $E[s]$ and have outcome ∞. Otherwise, the outcome is w.

Verification. Let TP be the true path, i.e. the leftmost path of nodes visited at infinitely many stages. We prove by induction on n that $TP \upharpoonright n$ succeeds to satisfy its requirement.

Lemma 9. *There is a stage s_n such that at stage $s > s_n$ the strategy $TP \upharpoonright n$ is not initialized any longer.*

Proof. The only issue is that an S strategy $\sigma \preceq TP$ may have outcome w but still initialize strategies below it. It does so only at expansionary stages in response to some number with defined g-value entering A, or when

enumerating a number with defined f-value into A. Since σ will not define further g or f values until all such numbers have been dealt with and this happens only at expansionary stages, if σ repeats this action infinitely often, it follows that σ has true outcome ∞. □

Lemma 10. *The functional* Γ *is total and satisfies the* H_e *requirements.*

Proof. First we note that ultimately Γ's values are defined by the H_e-strategy along the true path $\rho = TP \upharpoonright n$. This is because if ρ' is not visited after stage s it will not define any values for Γ and if ρ'' is initialized at stage s it will pick its parameter t_0 anew and that will have higher value. If some $\rho'' >_L TP$ defines $\Gamma^A(x,t) = 1$ then it will be for $t > t_\rho$ and because there was an increase in the domain of φ_x between consecutive ρ''-visits. It follows that when ρ is visited again, this increase will be observed by ρ and it will redefine $\Gamma^A(x,t)$ according to its own plan. In particular if ρ has true outcome w then φ_x is partial and eventually any value defined by any strategy for $\Gamma^A(x,t)$ will be 0 and whenever ρ is visited, such a value will be defined without an increase in the use. If on the other hand ρ has the infinite outcome, i.e. φ_x is total , then ρ will eventually define $\Gamma^A(x,t) = 1$ for all $t > t_0$. □

Lemma 11. *Every* S_e *requirement is satisfied*

Proof. Fix the S_e-strategy σ along the true path. If σ has true outcome w then σ has only finitely many expansionary stages and some disagreement between the set $m_e(W_e)$ and A is observed at every further stage. It follows that W_e is not reducible to A via m_e and/or A is not reducible to W_e via n_e. Otherwise, if there are infinitely many expansionary stages then we ensure that g_σ and f_σ are total functions and since σ is the only strategy that can enumerate $g_\sigma(x)$ in A, it follows that $x \in W_e$ if and only if $g_\sigma(x) \in A$. Furthermore, anytime that $f_\sigma(x)$ enters W_e, σ will enumerate x into A so $x \in A$ if and only if $f_\sigma(x) \in W_e$. □

Lemma 12. *Every* I *requirement is satisfied*

Proof. Suppose that $\tau = TP \upharpoonright n$ is the I-strategy on the true path. If τ never sees a believable computation then its true outcome is w and $\Phi_e^A(n_\tau) \neq 1$. Indeed, if $\Phi_e^A(n_\tau) = 1$ then there is some fixed initial segment $A \upharpoonright M$ that is used in this computation. We claim that at a large enough stage $s > s_n$ this computation will be believable. Indeed, as argued above every time

we visit $\rho^\frown\infty \preceq \tau$ for some H-strategy ρ it redefines the uses of its (x, t_ρ) to be so large that it accounts for all protected numbers, and it always picks the largest possible number to enumerate into A. Thus eventually the smallest number it extracts will be larger than M. So if the computation has appeared at stage $s > s_n$, it may seem unbelievable to τ only because of a marker $g(x)$ or $f(z)$ defined by some S-strategy σ with $\sigma^\frown\infty \preceq \tau$ such that $m_\sigma(x, s)$ or $m_\sigma(f(z), s)$ may still enter A. However, since (σ, x) and $(\sigma, f(z))$ are protected by all H-strategies from now on, if $m_\sigma(x, s), m_\sigma(f(z), s) \notin A_s$, they will never be added to A by an H-strategy. They may be added to A by an S-strategy σ' but only if $m_\sigma(x, s) = g_{\sigma'}(y)$ or $m_\sigma(f(z), s) = g_{\sigma'}(y)$ for some y. This means that σ' is responding to $m_{\sigma'}(y, s)$ entering A, which again won't happen at further stages due to H activity as $g_{\sigma'}(y)$ must be already defined at this stage. So unless this chain reaction is already set off at this stage, no H-strategy can set it off at further stages. By the next τ-visit, we have that the effects of the chain reaction have been cleared and no numbers less than M can ever be added to A by higher priority strategies. Thus τ would see the computation as believable, contrary to our assumption.

On the other hand, if τ ever sees a believable computation, then this computation will be preserved. By definition, it will not be injured by higher priority strategies. And lower priority strategies are in initial state, so if visited from now on, they will select t_0 or g-values preserving the initial segment of A that τ cares about. □

□

3 Topped degrees

Let $\leq_s$ and $\leq_w$ be a pair of a stronger and weaker reducibilities. If a w-degree is not s-contiguous, we may wonder about the $\leq_s$-order of the c.e. s-degrees inside it. As we saw in the introduction, of specific practical importance are the s-topped degrees, degrees that contain a largest s-degree. In this section, we examine this notion.

We start by examining the *m-topped* Q-degrees more carefully. We will try to understand how these degrees are distributed within the structure of the Q-degrees. For this we need the notion of a *semi-low* c.e. set.

Definition 13. A c.e. set A is *semi-low* if the set $\{e : W_e \cap \overline{A} \neq \emptyset\}$ is Δ^0_2.

Note that every low set is semi-low, however, being semi-low does not imply having a weak jump. In fact, Soare [27] proved that there are semi-low sets in every c.e. Turing degree.

We show that semi-lowness is an obstacle to being m-topped. The following theorem tells us more about the distribution of m-contiguous Q-degrees, namely that they must be non-low.

Theorem 14. *If A is a non-computable semi-low c.e. set then the Q-degree of A is not m-topped.*

Proof. Fix a semi-low set A and a Δ^0_2 approximation to the set $C = \{e : W_e \cap \overline{A} \neq \emptyset\}$. We will build a c.e. set B so that $B \leq_Q A$ via q but $B \not\leq_m A$. We need to satisfy the following requirements:

$$R_e : \text{if } \varphi_e \text{ is total then } (\exists x)[x \in B \not\leftrightarrow \varphi_e(x) \in A].$$

Construction. To every requirement R_e at stage s we have appointed witnesses $x_0, \ldots x_k$ with uses $q(x_i, s)$ covering some initial interval of $\overline{A}$. At stage $s+1$ we visit each R_e requirement in turn. We check whether one of the appointed witnesses is permitted to enter B at stage s. Suppose $q(x_i, s)$ has entered A at this stage. By the recursion theorem let $h(e)$ be a computable function such that $W_{h(e)}$ is the c.e. set of witnesses appointed by this strategy that are realized and permitted to enter A. If $h(e) \notin C_s$ then we wait for confirmation that $\varphi_e(x_i)$ is defined and in A: we speed up the approximation to φ_e and A to look for a stage $t > s$ when both of these fact are true. If we do find a stage t like that, we declare that this requirement is permanently satisfied by $x_i \notin B$ and so at further stages s' all we need to do is redefine $q(x_i, s')$ if it enters A to the next number that is not in A currently. This will happen finitely many times, because A is not computable. We do so for every permitted witness from the ones available.

While we are looking for t we also look ahead in the approximation to C to check whether $C(h(e))$ changes its mind to 1. If this does happen we enumerate all permitted witnesses x_i in B. Now we have that $x_i \in B$ but according to C we have that $\varphi_e(x_i) \notin A$.

If $C(h(e)) = 0$ but all witnesses defined so far that were permitted were enumerated in B (because C lied to us) then we define x_{k+1} to be a new number targeted for B with $q(x_{k+1}, s+1)$ set to the next available number in $\overline{A}$.

Verification. To see that this works, fix e and assume that φ_e is total. Towards a contradiction suppose that $x \in B$ if and only if $\varphi_e(x) \in A$. Let s be such that $C(h(e))[s]$ has already reached a limit. If this limit is 1 then by semi-lowness we know that some permitted realized witness defined by this strategy is in $\overline{A}$. The strategy will enumerate in B every permitted witness. Indeed, if x_i is permitted at stage s then either $C_s(h(e)) = 1$ and x_i enters B or else $C_s(h(e)) = 0$ in which case we start looking for confirmation. We will either get confirmation and diagonalize forever, or see that C flips back to 1. By assumption the latter happens and so x_i enters B. So there will be a witness x in B that such that $\varphi_e(x)$ never enters A. If, on the other hand, the limit is 0 then starting at stage s we will never again enumerate any witness in B, we will define more and more witnesses, but then since A is not computable, at least one should later on be permitted. This contradicts our assumption. □

We show that there is a non-low c.e. Turing degree that contains no m-topped Q- or sQ-degree.

Theorem 15. *There is a non-low c.e. Turing degree* **a** *which has no m-topped $(s)Q$-degree.*

Proof. We build A and a partial computable functional Γ^A to meet the non-lowness requirements

$$P_e : \exists x \Gamma^A(x) \neq \lim_s \varphi_e(x, s).$$

Essentially, we are building Γ^A to be the jump of A, so in order for A to be low, the opponent has to demonstrate that $\mathbf{0'}$ can compute A'. By the limit lemma, this means there has to be a total computable function whose limit is Γ^A, if A is to be low. Then A is not low if there is no such function, that is, if all the P_e-requirements are met. Here we can regard $\varphi_e(x, s)$ as primitive recursive, so we don't need to worry about $\varphi_e(x, n)[s]$, and can at stage s only look at $\varphi_e(x, s)$.

Additionally, we need any c.e. set W_e which is in the same Turing degree as A to not have an m-topped Q-degree, and so we meet the requirements

$$R_e : [\Theta_e^A = W_e \wedge \Psi_e^{W_e} = A] \to [X_e \leq_Q W_e \wedge \forall i(R_{e,i})], \text{ where}$$

$$R_{e,i} : \neg(X_e \leq_m W_e \text{ via } f_i).$$

We use lowercase letters to denote the uses, so for instance $\theta_e(x)$ is the use of the computation $\Theta_e^A(x)$. We use a tree and meet R_e at nodes τ during e-expansionary stages, where we build the set X_e and the Q-reduction m_e witnessing that $X_e \leq_Q W_e$. At τ we have two outcomes, $\infty <_L f$. The tree is sculpted so that we have the $R_{\tau,i}$-nodes σ devoted to meeting $R_{e,i}$ below $\tau^\frown\infty$. At σ we have two outcomes $1 <_L 0$ with 1 meaning we have diagonalized against f_i. We also interweave the P_e requirements, and at nodes ρ devoted to meeting P_e we have two outcomes, $\infty <_L f$, as described in the basic strategy below.

Basic Strategies. To meet the P_e requirements, we pick a follower x and keep $\Gamma^A(x)\uparrow$ until we see a stage where $\varphi_e(x,s) = 0$. Then declare $\Gamma^A(x)[s+1]\downarrow$ with a big use $\gamma(x)[s+1]$ (for example $\gamma(x)[s+1] = s+1$ or a fresh big number). Continue until we see a stage t where $\varphi_e(x,t) = 1$. Then we enumerate $\gamma(x)[t]$ into A_{t+1} and make $\Gamma^A(x)[t+1]\uparrow$. Repeat as required. Now if $\lim_s \varphi_e(x,s)$ exists then $\varphi_e(x,s)$ eventually has to stop changing between 0 and 1, and so P_e succeeds by acting once more and flipping one time more than $\varphi_e(x,s)$. This is the finite outcome f for P_e. Otherwise, if each time that P_e changes $\Gamma^A(x)$, $\varphi_e(x,s)$ changes to match, then P_e will keep flipping the output of $\Gamma^A(x)$ forever and so will $\varphi_e(x,s)$. In this case P_e acts infinitely many times, so this is the infinite outcome of P_e. In the infinite outcome, P_e will succeed because $\lim_s \varphi_e(x,s)$ doesn't exist.

For the R_e and $R_{e,i}$ requirements, we monitor the length of agreement $\ell(e,s) = \max\{x : \forall y \leq x[(\Psi^{W_e} \upharpoonright y = A \upharpoonright y) \wedge (\Theta^A \upharpoonright \psi(y) = W_e \upharpoonright \psi(y))][s]\}$. For the sake of $R_{e,i}$, at an e-expansionary stage we will pick an *agitator* z targeted for A which is fresh. The idea is that we will use this z to set up a scenario where we can diagonalize against f_i. Since we do this at an e-expansionary stage, z exceeds all computations seen so far. Let p_1 be the largest W_e-use seen below $\ell(e,s)$. That is, the largest W_e-use out of any computation that is below the current length of agreement. By our choice of z, we must have that $z > p_1$. Now we wait for a stage t where $\ell(e,t) > z$. If this never happens then R_e is satisfied because $W_e \not\equiv_T A$. Otherwise, let p_2 be the largest W_e-use of any computation below $\ell(e,t)$. Notice that if we were to put z into A after stage t, whilst also keeping $A \upharpoonright \theta_e(p_1)$ unchanged (that is, freeze A up to the A-use of the largest W_e-use, which forces $W_e \upharpoonright p_1$ to remain unchanged, in turn forcing $A \upharpoonright \ell(e,s)$ to remain unchanged), then some w must enter W_e between p_1 and p_2 in order for the length of agreement to recover. However, we do not put z into A just yet.

First, at this stage t we attend to $X_e \leq_Q W_e$, extending the definition of a Q-like function m_e. For each $w \in (p_1, p_2] \cap \overline{W_{e,t}}$ find a fresh number x_w and define $m_e(x_w, t) = w$. That is, x_w can enter X_e should w enter W_e.

Now we restrain $A \upharpoonright \theta_e(p_2)$ (so that A's computation Θ_e^A of $W_e \upharpoonright p_2$ remains unchanged) and wait for an e-expansionary stage $u > t$ such that for all w and x_w, we see $f_i(x_w) \downarrow [u]$. Assuming that f_i is total and hence a potential candidate for witnessing an m-reduction, such a stage u exists. Note that because of our restraint on A, we have that $W_{e,u} \upharpoonright p_2 = W_{e,t} \upharpoonright p_2$, so all the w's are still not in $W_{e,u}$. If for any x_w we have that $f_i(x_w) \in W_{e,u}$, then we don't need to do anything except keeping this particular x_w out of X_e. Now f_i is wrong on $x_w \notin X_e$ and has been diagonalized.

Otherwise, we know that for all w, $f_i(x_w) \notin W_{e,u}$. Now we will attack this by adding z to A_{u+1} and otherwise restraining $A \upharpoonright \theta_e(p_2)$, causing some w to enter W_e between p_1 and p_2, at the next e-expansionary stage v. There are two cases:

- Case 1. For some x_w, $f_i(x_w)$ has entered W_e. Then we again simply keep x_w out of X_e, and so diagonalize. If $f_i(x_w) = m_e(x_w, t)$ then we retarget $m_e(x_w, v)$ to something small that is not in W_e.
- Case 2. For some w entering $(p_1, p_2]$, $f_i(x_w)$ does not enter W_e. Then we can win by putting x_w into $X_{e,v+1}$ and otherwise restraining $A \upharpoonright v$. Now since the stage number v bounds any computation by convention, A's computation of $W_e(f_i(x_w))$ does not change, and so for the length $\ell(e, s)$ to go to infinity, $f_i(x_w)$ can never enter W_e. Thus the disagreement $x_w \in X_e$ (and $w \in W_e$) while $f_i(x_w) \notin W_e$ is preserved forever.

Interactions between requirements. The P_e requirements do not interfere with each other, since when the use $\gamma(x)[s]$ is declared, it is bigger than any computations and uses at stage s, in particular bigger than any use $\gamma(x')$ declared by any other P requirement. The R_e requirements also do not interfere with one another, since a higher priority R_e requirement will simply initialize a lower priority $R_{e'}$ requirement whenever R_e acts, and if the lower priority $R_{e'}$ requirement acts, it will have picked its agitator z to be larger than anything seen before, and in particular larger than any restraints set by R_e.

The only complicated interactions that can arise are between R and P

requirements. Let τ be the mother of $R_{e,i}$ (that is, R_e is assigned to τ), let $R_{e,i}$ be assigned to node σ, and let P_k be assigned to node ρ. Recall that τ has outcomes $\infty <_L f$, as does ρ.

If τ is below $\rho^\frown\infty$ then it does not believe a computation until $\gamma(x)$ has cleared the use of that computation. That is, P_ρ will not affect this computation should it enumerate the use $\gamma(x)[s]$ into A. If τ is below $\rho^\frown f$ then P_ρ initializes R_τ each time it acts.

If ρ is below $\tau^\frown\infty$ and also below σ, then R_σ initializes P_ρ each time it receives attention.

Finally, we have the configuration $\tau^\frown\infty \subseteq \rho^\frown\infty \subseteq \sigma$. Now σ is only reached at $\tau^\frown\infty$-stages that are also $\rho^\frown\infty$-stages, so stages when P_ρ acts. Whenever P_ρ enumerates an element into A, we will end the stage without attending to any other requirements. Thus at the stage s when R_σ is first attended to, P_ρ has just declared a big use $\gamma(x)[s]$. After this, R_σ picks z to be a fresh number, in particular, larger than $\gamma(x)[s]$. Then the next time P_ρ is visited it enumerates $\gamma(x)[s]$ into A and ends the stage. At the next $\rho^\frown\infty$ stage t, P_ρ declares a new large use $\gamma(x)[t]$. Now R_σ is attended to again. If $\ell(e,t) > z$ then R_σ will begin its attack. Notice now that $\gamma(x)[t]$ is too big to affect R_σ's attack, because it has been picked to be bigger than everything *at stage* t, so must be larger than $\theta_e(p_2)$. This means that now P_ρ's actions will not affect R_σ's attack. If $\ell(e,t) \leq z$ then R_σ will wait, and at the stage t' when $\ell(e,t') > z$ we will have exactly the same setup: the $\gamma(x)[t']$ that P_ρ declared will be too large to affect R_σ's attack.

Construction. At stage s, compute TP_s. Attend to requirements along TP_s as described above.

Verification. Let the true path TP be the leftmost path visited infinitely often.

Lemma 16. *Each $R_{e,i}$ on the true path acts finitely often.*

Proof. We will show this by induction. Suppose $R_{e,i}$ is assigned to node $\sigma \subset TP$. Go to a stage s_0 after which TP_s, $s \geq s_0$ is not to the left of σ. This means that by stage s_0, any higher priority R or P requirements whose finite outcome is on the true path have all finished acting, because the infinite outcome of these requirements is not visited again. Suppose also that by stage s_0, any $R_{e',i'}$ on the true path above σ has finished acting (these only act finitely often by the inductive hypothesis). Now after stage

s_0, R_σ is not initialized again. This is because for any P_ρ above R_σ that act after stage s_0, we have that $\rho^\frown\infty \subseteq \sigma$. Then as discussed in the interactions between requirements, the actions of P_ρ do not inhibit the action of R_σ. Now R_σ can proceed as in its basic strategy, and will act at most finitely many times: when picking the agitator z, when picking the elements x_w and defining $m_e(x_w, t)$, and finally (if needed) when acting by one of the cases in the basic strategy. □

Lemma 17. *Every P_e requirement is met.*

Proof. Suppose ρ is on the true path and that P_e is assigned to ρ. There are only finitely many R_σ above P_ρ, each of which receive attention and act only finitely often, and so P_ρ is only initialized finitely many times (and any $P_{\rho'}$ above P_ρ do not interfere with P_ρ). After P_ρ has been initialized for the last time, it is free to pick a witness x and act as in the basic strategy. Then P_ρ is visited infinitely many times and is met either by changing once more than $\varphi_e(x, s)$, or by changing infinitely many times and making $\lim_s \varphi_e(x, s)$ be undefined. □

Lemma 18. *Every R_e requirement is met.*

Proof. Let $\tau \subset TP$ be the node on the true path that R_e is assigned to. If $\tau^\frown f \subset TP$, then R_e is met: the finite outcome means that the length of agreement $\ell(e, s)$ gets stuck forever, so $W_e \not\equiv_T A$.

Suppose $\tau^\frown\infty \subset TP$. That is, there are infinitely many τ-expansionary stages, i.e. $\ell(e, s)$ goes to infinity. Go to a stage s_0 after which TP_s, $s \geq s_0$ is not to the left of τ. That is, by stage s_0 every higher priority R and P requirement whose finite outcome is on the true path have finished acting, so R_τ is never initialized again.

We show by induction that every $R_{e,i}$ is met. Let $\sigma \subset TP$ be the node on the true path that $R_{e,i}$ is assigned to. Go to the first stage $s_1 \geq s_0$ such that any $R_{e',i'}$ above σ has finished acting, and TP_s is never again to the left of σ. Such a stage exists by Lemma 16. Now after stage s_1, R_σ is never initialized again. At the next $\tau^\frown\infty$ stage s, R_σ picks an agitator z targeted for A, and this agitator is final. At stage t, z is bigger than anything seen thus far, including any $\gamma(x)[s]$ declared by higher priority P requirements whose infinite outcome is above σ. At the next $\tau^\frown\infty$ stage t where $\ell(e, t) > z$, which is also a $\rho^\frown\infty$ stage for P_ρ on the true path above σ, R_σ restrains $A \upharpoonright \theta_e(p_2)$ and assigns a new number x_w targeted for X_e for each $w \in (p_1, p_2] \cap \overline{W_{e,t}}$. As

discussed, all $\gamma(x)[s]$ elements declared by P_ρ above have been enumerated and the length of agreement has since recovered. The new elements $\gamma(x)[t]$ are fresh, in particular bigger than p_2. We define $m_e(x_w, t) = w$. Because all numbers that may be enumerated into A are now larger than $\theta_e(p_2)$, the restraint on $A \upharpoonright \theta_e(p_2)$ is respected. Now if f_i is not total and does not halt for some x_w, then R_σ never acts again, and $R_{e,i}$ is met. Otherwise, R_σ can continue as per the basic strategy, because now any element that may be enumerated into A is too large to affect the strategy. In each of the remaining cases, R_σ successfully diagonalizes against f_i, and thus $R_{e,i}$ is met.

Now we check that the set X_e constructed by meeting all the $R_{e,i}$ requirements is indeed Q-below W_e. To see that m_e is total, notice that each element is eventually assigned as x_w for some w. At the stage s where this happens, we defined $m_e(x_w, s) = w$, and left $m_e(x_w, t+1) = m_e(x_w, t)$ for all $t \geq s$ unless otherwise specified. If nothing further happens to x_w and w then this will be correct as neither x_w nor w will have entered their respective sets. Now suppose w enters W_e at some stage $t > s$. It is possible this happens because the requirement R_σ which defined $m_e(x_w, s)$ has been initialized by some higher priority requirement, removing a restraint on A which in turn allowed w to be enumerated into W_e. In this case, implicit in our construction was that R_τ ensures the consistency of m_e at expansionary stages, so $m_e(x_w, t)$ is retargeted to something small not in W_e (equally we could just enumerate x_w into X_e). Otherwise (if R_σ has not been initialized), w entered as a result of R_σ's attack, that is, R_σ enumerated its agitator z into A. Then the definition of $m_e(x_w, t)$ was updated depending on which case R_σ was in (either R_σ enumerated x_w into X_e, or it retargeted $m_e(x_w, t)$ to something small not in W_e). In either case, we end up with $x_w \in X_e$ if and only if $m_e(x_w) = \lim_s m_e(x_w, s) \in W_e$. Thus $X_e \leq_Q W_e$ as witnessed by m_e.

Putting this all together, if the length of agreement goes to infinity, so $W_e \equiv_T A$, then the version of R_e at node τ on the true path successfully builds $X_e \leq_Q W_e$ and all $R_{e,i}$ (at nodes σ below τ) are met, hence R_e is satisfied. □

Since all the R_e and P_e requirements are met, we have constructed a set A with the desired properties. Notice that the Q-reduction m_e is also an sQ-reduction since $m_e(x_w) \leq w$ for every x_w. □

We next consider the notion of being topped for other pairs of a strong reducibility and a weak reducibility. Downey, LaForte and Nies [9] established that semi-recursive sets play the role of Q-tops for the c.e. Turing degrees.

Definition 19. A set A is *semi-recursive* if A is a left cut in some computable linear ordering on ω.

We see that semirecursiveness plays a similar role for the pair sQ- and wtt-reducibility. Jockusch [16] showed every c.e. wtt-degree has a c.e. semirecursive set, thus every wtt-degree is sQ-topped.

Proposition 20. *If* $\mathbf{a}$ *is a c.e. wtt-degree and* $A \in \mathbf{a}$ *is a c.e. semirecursive set, then* $W \leq_{wtt} A$ *implies that* $W \leq_{sQ} A$*, for any c.e. set* W*.*

Proof. Suppose A is semirecursive via $f(x, y)$, and that $\Gamma^A = W$ is the wtt-procedure witnessing $W \leq_{wtt} A$ with use function γ. Define the length of agreement $\ell(s) = \max\{x : \forall y \leq x,\ \Gamma^{A_s}(y) = W_s(y)\}$ and maximum length of agreement $m\ell(s) = \max\{\ell(t) : t < s\}$. We call a stage s *expansionary* if $\ell(s) > m\ell(s)$.

Suppose that the wtt-procedure Γ has been accelerated so that every stage is expansionary. We define a Q-like function $m(x, s)$ for $x \leq \ell(s)$ as follows.

Suppose at stage s we are defining m on x for the first time and $\Gamma^{A_s}(x) = 0$ (say if x is below the length of agreement for the first time), or that $m(x, s-1) \in A_s$ but $\Gamma^{A_s}(x) = 0$. In order for $\Gamma^{A_s}(x)$ to change and thus for x to enter W, (at least) one element below the use $\gamma(x)$ must enter A. Compute $f(x, y)$ for $x, y \in \mathbb{N} \upharpoonright \gamma(x) \setminus A_s$ and use this to determine the most preferred element z, which must enter A if any other element(s) below $\gamma(x)$ enter. This element exists because whenever $f(x, y) = x$, x is preferred over y (and conversely if $f(x, y) = y$), because if y enters A, x must also enter A. Now set $m(x, s) = z$.

If instead $m(x, s-1) \notin A_s$ is defined and $\Gamma^{A_s}(x) = 0$, set $m(x, s) = m(x, s-1)$.

Otherwise, $\Gamma^{A_s}(x) = 1$ (so $x \in W_s$). If this is the first time we are defining m on x, let $m(x, s) = x_0$ for some fixed $x_0 \in A$. If not, $m(x, s-1)$ is defined, so set $m(x, s) = m(x, s-1)$. Now either $m(x, s-1) \in A_s$ already, or some other small number entered A_s and caused $\Gamma^{A_s}(x)$ to change, but

$m(x, s-1)$ hasn't entered A_s. However, because $m(x, s-1)$ is a preferred element, it must enter A at some stage t and we will have $\Gamma_t^A(x) = \Gamma_s^A(x) = 1$, since every stage is expansionary.

Then $m(x) = \lim_s m(x, s)$ exists and $m(x) \in A$ if and only if $x \in W$, and furthermore $m(x) \leq \max\{\gamma(x), x_0\}$. Thus we have an sQ-reduction $W \leq_{sQ} A$. □

4 Minimal pairs and Capping

In this section we examine two related notions: minimal pairs and being cappable, i.e. being half of a minimal pair.

Recall that a pair of sets A and B is a minimal pair with respect to a reducibility $\leq_r$ if whenever $X \leq_r A$ and $X \leq_r B$, we have that X is computable. Note that a minimal pair with respect to a weak reducibility is automatically a minimal pair with respect to any stronger reducibility. The reverse is not true. In fact we can have a minimal pair with respect to a strong reducibility where the degrees of the pair with respect to a weak reducibility are the same. We examine in which cases this is possible.

Downey and Stob showed in 1986 that there is a minimal pair of *wtt*-degrees within the same Turing degree [13]. The proof (which uses a pinball machine) can be seen to give the same result but with the sQ- and Q-degrees. That is, there is a minimal pair of sQ-degrees within the same Q-degree. Downey, LaForte, and Nies [9] construct a Q-minimal pair A and B within the same Turing degree. The following theorem implies that we cannot have a minimal pair of Q-degrees (or sQ-degrees) within the same *wtt*-degree.

Theorem 21. *If $A \leq_{wtt} B$ are c.e., then there exists $D \equiv_{wtt} A$ with $D \leq_{sQ} A, B$. Hence if $A \equiv_{wtt} B$ then if the infimum of A and B in the Q-degrees or sQ-degrees exists, it will have the same wtt-degree as A.*

Proof. Let $\Gamma^B = A$ be the *wtt*-procedure witnessing that $A \leq_{wtt} B$ with use function γ, and assume this is accelerated so that every stage is expansionary via the length of agreement function $\ell(s)$. We construct the c.e. set D as follows.

Construction. For each i we define a set $P_{i,j} = \{x_{i,j} : 0 \leq j \leq \gamma(i)\}$. At stage $s+1$ we put $x_{i,j}$ into D if one of the following occurs.

1. Both $i \in A_{s+1} \setminus A_s$ and $j \in B_{s+1} \setminus B_s$. That is, i and j both enter between stages s and $s+1$, simultaneously changing A and the computation of Γ.

2. $i \in A_s$, $j \in B_{s+1} \setminus B_s$ and $\Gamma^{B_s}(i) \neq \Gamma^{B_{s+1}}(i) = 1$ for the first time. That is, i was already in A at stage s but $\Gamma^{B_s}(i) = 0$. Note that this means $i > \ell(s)$. Then $j \leq \gamma(i)$ entered B_{s+1} making the computation change to match. If there is no $i' < i$ causing a disagreement, now $i \leq \ell(s+1)$.

3. $i \in A_{s+1} \setminus A_s$ but $B_s \upharpoonright \gamma(i) = B_{s+1} \upharpoonright \gamma(i)$, and $\Gamma^{B_s}(i) = 1$. Further, $j \leq \gamma(i)$ is the last such j to have entered B. That is, i enters making the computations match, but B did not change below the use $\gamma(i)$ since the computation was already correct. Again we have $i > \ell(s)$ and if there is no $i' < i$ causing a disagreement, then $i \leq \ell(s+1)$.

Verification. We claim $D \leq_{sQ} A, B$. To see this, construct Q-like functions m and n, with m targeted for A and n for B.

Define $m(x_{i,j}, 0) = i$ and fix $a \notin A$. Unless otherwise stated, we keep $m(z, s+1) = m(z, s)$.

Suppose $i \in A_{s+1} \setminus A_s$. If $j \in B_{s+1} \setminus B_s$, define $m(x_{i,k}, s+1) = a$ for each $k \neq j$, and keep $m(x_{i,j}, s+1) = m(x_{i,j}, s) = i$. If instead $B_{s+1} \upharpoonright \gamma(i) = B_s \upharpoonright \gamma(i)$ and $\Gamma^{B_s}(i) = 1$, look at the enumeration of B up to stage $s+1$ and let j be the last $j \leq \gamma(i)$ to enter B during this time. Let $m(x_{i,k}, s+1) = a$ for each $k \neq j$, and keep $m(x_{i,j}, s+1) = m(x_{i,j}, s) = i$.

Now suppose $j \leq \gamma(i)$ enters B at stage $s+1$ and i is already in A_s. If $\Gamma^{B_{s+1}}(i) = 0$, define $m(x_{i,j}, s+1) = a$. If $\Gamma^{B_{s+1}}(i) = 1 \neq \Gamma^{B_s}$ and $m(x_{i,k}, s) = k$, define $m(x_{i,k}, s+1) = a$ for each $k \neq j$ and keep $m(x_{i,j}, s+1) = m(x_{i,j}, s) = i$.

It is not difficult to verify that $x_{i,j} \in D$ if and only if $\lim_s m(x_{i,j}, s) = m(x_{i,j}) \in A$. Further, $m(x_{i,j}) \leq \max\{i, a\}$ and so this is an sQ-reduction, thus $D \leq_{sQ} A$.

Now for our second reduction, define $n(x_{i,j}, 0) = j$ and fix $b \notin B$. Unless otherwise stated, we keep $n(z, s+1) = n(z, s)$.

Suppose $j \in B_{s+1} \setminus B_s$. If $i \in A_{s+1} \setminus A_s$, let $n(x_{i',j}, s+1) = b$ for all $i' \neq i$, and keep $n(x_{i,j}, s+1) = n(x_{i,j}, s) = j$. If $i \notin A_s$, define $n(x_{i,j}, s+1) = b$. Finally if $i \in A_s$ and $\Gamma^{B_{s+1}}(i) = 1 \neq \Gamma^{B_s}(i)$ and $n(x_{i,k}, s) = k$, let

$n(x_{i,k}, s+1) = b$ for all $k \neq j$ and keep $n(x_{i,j}, s+1) = j$.

Then $x_{i,j} \in D$ if and only if $\lim_s n(x_{i,j}, s) = n(x_{i,j}) \in B$, and $n(x_{i,j}) \leq \max\{\gamma(i), b\}$. Thus we have an sQ-reduction $D \leq_{sQ} B$.

Finally, since $D \leq_{sQ} A$ we have $D \leq_{wtt} A$, so to see that $A \equiv_{wtt} D$ we need to show that $A \leq_{wtt} D$. To decide if $i \in A$, wait for a stage s where $i \leq \ell(s)$ and $D_s \upharpoonright \max\{x : x \in P_{i,j}\} = D \upharpoonright \max\{x : x \in P_{i,j}\}$. Then $i \in A$ if and only if $i \in A_s$, since every stage being expansionary implies that if i enters A_t at a later stage t, some $j \leq \gamma(i)$ must also enter B at the same stage. □

Even though being a minimal pair with respect to a strong reducibility does not imply being one with respect to a weaker reducibility, in some cases the property of being half of a minimal pair transfers from a stronger to a weaker reducibility. Ambos-Spies et al. [2] prove that the c.e. Turing degrees of *promptly simple sets* are exactly the non-cappable c.e. Turing degrees. As a step in their proof they give a characterization of when a c.e. set has promptly simple Turing degree. Putting these two results together we have the following:

Theorem 22 (Ambos-Spies et al. [2])**.** *A c.e. set A is non-cappable in the Turing degrees if and only if there is a non-decreasing computable function p and a computable enumeration $\{A_s\}_{s<\omega}$ to A such that for every infinite c.e. W there is an s and an x such that x is enumerated in W at stage s (in some standard uniform enumeration of all the c.e. sets) and $A_s \upharpoonright x \neq A_{p(s)} \upharpoonright x$.*

It is not hard to see that in the above characterization there are infinitely many x and s satisfying the prompt property for every infinite c.e. set W. We use this to show the following.

Theorem 23. *If A is a c.e. set that is half of a minimal pair in the sQ-degrees then A is half of a minimal pair in the Turing degrees.*

Proof. Fix A that is not half of a Turing minimal pair, i.e. A is non-cappable, and let p be the computable function from Theorem 22. Let A be promptly simple via p and let B be a given non-computable c.e. set. We build a set $Z \leq_{sQ} A, B$ with sQ-reductions m and q respectively, to meet requirements

$$R_e : Z \neq \overline{W_e}.$$

Basic Strategy. To meet the requirement R_e we proceed in cycles $n = n_e$. At cycle n which begins at stage s_n, say, we define followers $z_{i,n} \notin Z_{s_n}$ for all $i < n$ such that $i \notin A_{s_n}$. We set $m(z_{i,n}, s_n) = i$ and $q(z_{i,n}, s_n) = n$. Once all these followers have been realized (have entered W_e) at stage s_{n+1}, we define new followers $z_{i,n+1} \notin Z$ for all $i < n + 1$. We continue this way defining groups of followers $\{z_{i,n}\}_{i<n}$ so long as R_e is not yet met. To meet R_e we either find a follower that is never realized or we need to put a single realized $z_{i,n}$ into Z. In order for Z to be sQ-below A and B, we can only put $z_{i,n}$ in Z if both $m(z_{i,n})$ and $q(z_{i,n})$ enter A and B respectively. Here we use the promptly simple properties of the degree of A. Since B is non-computable, if every follower is realized then there must be infinitely many n such that n enters B after the followers from cycle n enter W_e. When we see such an n at stage s, we enumerate it in an auxiliary c.e. set H. Now, by the properties of p and A and since H will be infinite we will eventually encounter an i so that $i \in A_{p(s)} \setminus A_s$. When we find it, we can enumerate $z_{i,n}$ in Z to satisfy the requirement.

Construction. Fix some $a \notin A$ and $b \notin B$. At stage s, run through substages e for $e \leq s$ as follows. If R_e is already met, $z^e_{i,n}$ is defined and not in Z, then if $n \in B_{s+1} \setminus B_s$, set $q(z^e_{i,n}, s + 1) = b$, and if $i \in A_{s+1} \setminus A_s$, set $m(z^e_{i,n}, s + 1) = a$. Otherwise if R_e is not yet met:

1. If this is the first stage when we activate R_e then set R_e's cycle to be $n_e = 0$.
2. If all n_e-followers are in $W_{e,s}$ then set $n_e = n_e + 1$.
3. If R_e does not have n_e-followers then assign new followers, $z^e_{i,n_e} \notin Z_s$ for all $i < n_e$.
4. Consider all followers $z^e_{i,n}$ where $i < n \leq n_e$ in turn. Unless otherwise stated we keep $m(z^e_{i,n}, s+1) = m(z^e_{i,n}, s)$ and $q(z^e_{i,n}, s+1) = q(z^e_{i,n}, s)$.
   - If $n \in B_{s+1} \setminus B_s$ and $i \in A_{p(s)+1} \setminus A_s$ then enumerate $z^e_{i,n}$ in Z. Declare this requirement satisfied.
   - If $n \in B_{s+1} \setminus B_s$ but $i \notin A_{p(s)+1} \setminus A_s$ then set $q(z^e_{i,n}, s + 1) = b$.
   - If $i \in A_{s+1} \setminus A_s$ but $n \notin B_{s+1} \setminus B_s$ then set $m(z^e_{i,n}, s + 1) = a$.
   - If $z^e_{i,n}$ is a new follower and $i \notin A_s$ and $n \notin B_s$ then set $m(z^e_{i,n}, s + 1) = i$ and $q(z^e_{i,n}, s + 1) = n$.
   - If $z^e_{i,n}$ is a new follower and either $i \in A_s$ or $n \in B_s$ then set $m(z^e_{i,n}, s + 1) = a$ and $q(z^e_{i,n}, s + 1) = b$.

Verification. Suppose R_e is not met. Then every follower $z^e_{i,n}$ that is assigned to R_e is eventually realized (enters W_e), because if $z^e_{i,n}$ is not realized then it is not in W_e and not in Z, and thus R_e is satisfied. So n_e grows unboundedly. Consider the c.e. set H of all numbers n such that n enters B after all n-followers are realized. If H is finite then B is computable because for $n > \max(H)$, to compute $B(n)$ all we need to do is wait for a stage s in the construction such that all s-followers are realized and then check $B_s(n)$. Thus H is infinite and by Theorem 22 we have that there is some s and n such that n enters B at stage $s+1$ and $A_{s+1} \upharpoonright n \neq A_{p(s+1)} \upharpoonright n$. So some $i < n$ enters A between stages $s+1$ and $p(s+1)$. By construction we will spot this i and enumerate the witness $z^e_{i,n}$ in Z.

Now we verify that m and q are valid sQ-reductions. Note that if $z^e_{i,n}$ enters Z then by construction we have that $m(z^e_{i,n}, s) = i$ and $q(z^e_{i,n}, s) = n$ and proof that $n \in B_{s+1} \setminus B_s$ and $i \in A_{p(s+1)} \setminus A_s$. In all other cases whenever we see a change in either $A(i)$ or $B(n)$ then we retarget m and q to safe numbers $a \notin A$ and $b \notin B$ and we never enumerate $z^e_{i,n}$ in Z. Finally, $m(z^e_{i,n}) \leq \max\{i, a\}$ and $q(z^e_{i,n}) \leq \max\{n, b\}$ and so these are sQ-reductions. □

5 One-types

In 1989 Ambos-Spies and Soare [3] proved that there are infinitely many one-types in the Turing degrees. They do this by proving that for $n \geq 2$, there are degrees that bound n degrees that pairwise form minimal pairs, but do not bound $n+1$ such degrees. We prove that the Q-degrees also have infinitely many 1-types, by using the following result to obtain Q-degrees with the same properties as the degrees that Ambos-Spies and Soare used for their result.

Theorem 24. *Suppose that* $\mathbf{b_1}, \mathbf{b_2}$ *are Q-degrees which form a Q-minimal pair. Then there are* $\mathbf{a_1}, \mathbf{a_2}$ *with* $\mathbf{0} <_T \mathbf{a_i} \leq_{sQ} \mathbf{b_i}$ *forming a T-minimal pair.*

Proof. Let B_1, B_2 be given c.e. sets that form a Q-minimal pair. We construct $A_i \leq_{sQ} B_i$ such that A_1, A_2 are a T-minimal pair.

We need to build $A_i \leq_{sQ} B_i$ to meet the following requirements

$$N_e : \Phi_e^{A_1} = \Phi_e^{A_2} = f \text{ implies } f \text{ is computable,}$$

$$P_e^i : A_i \neq \overline{W_e}.$$

Basic Strategies. We use a tree and meet the P_e^i requirements by a standard Friedberg strategy, made more complicated as we need to keep $A_i \leq_{sQ} B_i$. This is done by letting x enter A_i only if x enters B_i. That is, the sQ-reduction $A_i \leq_{sQ} B_i$ is witnessed simply by $m_i(x) = x$.

Further, P_e^i needs to respect higher priority negative requirements N_k at nodes τ where $\tau^\frown\infty$ is above P_e^i on the tree. The problem is that P_e^i may only get permission to put its witness x into A_i at stages that are not $\tau^\frown\infty$-stages. We need to ensure that P_e^i can put elements into A_i without destroying our computable definition of $f \upharpoonright n$.

To do this, instead of letting N_k freeze A_1 or A_2 as in a standard minimal pair argument (and thus prevent P_e^i from enumerating its witness), we will use the fact that B_1 and B_2 form a Q-minimal pair. We'd like to argue that if we put x into A_1 as it entered B_1, then no τ-injurious y can enter B_2 during the τ-gap between expansionary stages (having y enter B_2 would permit it to also enter A_2, which could then ruin the computability of f). So at τ we will build an auxiliary set $Z \leq_Q B_1, B_2$ and try to satisfy for all d,

$$N_{k,d} : Z \neq \overline{W_d}.$$

Now, we *cannot* meet all the $N_{k,d}$ since then there would be a non-computable $Z \leq_Q B_1, B_2$, contradicting that B_1 and B_2 are a Q-minimal pair. Thus we must get stuck on some d. That is, below $\tau^\frown\infty$ we have infinitely many sub-outcomes $d \in \omega$, given as a sequence $0 <_L 1 <_L 2 <_L \ldots$, and the leftmost one which we fail to meet will be the relevant d.

The approximation to the common value of $\Phi_k^{A_i}$ will be controlled by d, and each $\tau^\frown\infty^\frown d$ has its own incarnation.

Suppose we are at the d for which we fail to meet $N_{k,d}$. The requirement $N_{k,d}$ will work in cycles: at cycle n, first when $\ell(\tau, s) > n$ we allow $\tau^\frown\infty^\frown d$ to assert control of (restrain) $A_1 \upharpoonright \varphi_k^{A_1}(n, s)$ and $A_2 \upharpoonright \varphi_k^{A_2}(n, s)$. Then define fresh followers $z_{i,j}$ for $i \leq \varphi_k^{A_1}(n, s)$ and $j \leq \varphi_k^{A_2}(n, s)$. Once all of these are realized in that for all such i, j, $z_{i,j} \in W_{d,s}$, we will begin cycle $n + 1$ at the next $\tau^\frown\infty$ stage. While we are waiting for the followers to be realized, we go to outcome $d + 1$ and do the same (if some follower is never realized then $N_{k,d}$ is met, but we do not know this at any finite stage). When we start cycle $n + 1$ at outcome d, we remove the restraint on $A_1 \upharpoonright \varphi_k^{A_1}(n, s)$ and $A_2 \upharpoonright \varphi_k^{A_2}(n, s)$ and put a new restraint on A_1 that's just for numbers

above $\varphi_k^{A_1}(n,s)$ and below $\varphi_k^{A_1}(n+1,s)$, and similarly for A_2.

Then $N_{k,d}$ is allowed to put a follower $z_{i,j}$ into Z if there are two consecutive $\tau^\frown\infty$ stages $s_1 < s_2$, where i enters B_1 and j enters B_2. But then $N_{k,d}$ will be met, so assuming we are at the correct d this cannot happen.

Now versions of P_e^i living below $\tau^\frown\infty^\frown d$ will enumerate numbers x which are below the uses $\varphi_k^{A_1}(n,s)$ and $\varphi_k^{A_2}(n,s)$, where n has been treated and for which we have begun cycle $n+1$.

The idea is that as the length of agreement between $\Phi_k^{A_1}$ and $\Phi_k^{A_2}$ continues to increase, we will allow P_e^i below to assert control of more and more of the construction, in that as x which have been treated by our construction enter $B_i[s]$, we can immediately use them to meet P_e^i by putting them into $A_i[s]$.

The Priority Tree. Our tree has nodes σ assigned to P_e^i requirements with two outcomes, $0 <_L 1$. The outcome $\sigma^\frown 1$ means that some version of P_e^i has enumerated a follower into A_i. Nodes τ are assigned to requirements N_e, with two outcomes $\infty <_L w$, with ∞ meaning the length of agreement $\ell(\tau,s)$ goes to infinity and the finite ("wait") outcome w meaning the length of agreement gets stuck. Directly below $\tau^\frown\infty$ we have infinitely many outcomes $0 <_L 1 <_L 2 <_L \ldots$ with outcome $\tau^\frown\infty^\frown d$ assigned to requirement $N_{e,d}$ where N_e is the requirement assigned to τ. The tree is sculpted so that below each $\tau^\frown\infty^\frown d$ we have versions of each lower priority $P_{e'}^i$ (that is, some node σ below each $\tau^\frown\infty^\frown d$ is assigned to $P_{e'}^i$), and we also have a version of $P_{e'}^i$ below $\tau^\frown w$.

Construction. At stage s, compute TP_s. Act for any $N_{k,d}$ along TP_s as in the basic strategy, either by defining followers or enumerating a follower into the Z that $N_{k,d}$ is constructing. Requirements P_e^i along TP_s get as followers all the elements x that have been treated by all higher priority $N_{k,d}$ requirements (that is, $N_{k,d}$ has started a cycle $n+1$ where $\varphi_k^{A_i}(n,s) > x$). If some x enters B_i at stage s and x is a realized follower for some P_e^i along TP_s, then enumerate x into A_i.

Verification. Let the true path TP be the leftmost path visited infinitely often. First note that any P_e^i requirement acts at most once: when a follower has been enumerated by any version of P_e^i on the tree then P_e^i is met and no version of P_e^i ever acts again.

Lemma 25. *Every N_e requirement is met.*

Proof. Consider N_e at node $\tau \prec TP$. If $\tau^\frown w$ is on the true path then there are only finitely many τ-expansionary stages, so the length of agreement gets stuck, $\Phi_e^{A_1} \neq \Phi_e^{A_2}$ and N_e is met.

Now suppose $\tau^\frown\infty \prec TP$, and let d be so that $\tau^\frown\infty^\frown d$ is on the true path. For any $c < d$, $\tau^\frown\infty^\frown c$ is only visited finitely many times and then is never visited again, either because $N_{\tau,c}$ is met (some realized follower $z_{i,j}$ gets permission from both B_1 and B_2 and is enumerated into Z) or because $N_{\tau,c}$ gets stuck at some cycle n due to a follower $z_{i,j}$ never getting realized. Either way, $N_{\tau,c}$ has only restrained a finite amount of A_1 and A_2 (possibly none, if $N_{\tau,c}$ has been met), and any requirements not to the left of $N_{\tau,c}$ work above this finite restraint. Go to a stage s_0 large enough so TP_s for $s \geq s_0$ is never to the left of $\tau^\frown\infty^\frown d$, where $\tau^\frown\infty^\frown d \prec TP$. Note that this means any P_e^i requirement on a node σ above τ that will ever enumerate a follower, has already done so by stage s_0. Now the version of N_e at node τ is never initialized again, and we can believe the approximation to the common value of $\Phi_e^{A_i}$ that is controlled by d.

To compute $\Phi_e^{A_1} = \Phi_e^{A_2} = f$ up to some n, go to a $\tau^\frown\infty$-stage $s \geq s_0$ for which $N_{\tau^\frown d}$ is at a cycle greater than n. Now the length of agreement is above n and $\Phi_e^{A_1}(n)[s] = f(n)$. This is because $\tau^\frown\infty^\frown d$ is on the true path, so we know $N_{\tau,d}$ is never met, and from now on whenever the common approximation is updated at an expansionary stage, the length of agreement is above n. As described in the basic strategy, the common value is preserved and so the approximation given by $\tau^\frown\infty^\frown d$ is correct on n. Thus f is computable and N_e is met. □

Lemma 26. *Every P_e^i is met.*

Proof. Let $\sigma \prec TP$ be assigned to P_e^i. If $\sigma^\frown 1$ is on the true path, then some version of P_e^i enumerates a follower that has been realized, and so P_e^i is met. Otherwise, if $\sigma^\frown 0$ is on the true path then no version of P_e^i ever enumerates a follower. Let P_σ be the version of P_e^i at node σ. Now, P_σ uses as followers elements that have been treated by $N_{\tau^\frown\infty}$ above P_σ on the tree. Go to a stage s_0 after which TP_s for $s \geq s_0$ is never to the left of σ, so P_e^i does not get initialized again. If P_σ gets a follower that is never realized, then P_e^i is met as usual. The only consideration left is if every follower x that gets assigned to P_σ is either enumerated into B_i before x gets realized, or it gets realized but never gets permission from B_i to be enumerated. Note that P_σ gets more and more elements as followers, as these elements get treated by higher priority N_τ requirements. We claim that in this case B_i

is computable, and so this cannot happen. To compute $B_i(n)$, wait for n to be assigned as a follower for P_σ. We know this will happen at some stage s once n has been treated (unless n is enumerated into B_i before it is assigned as a follower, in which case we know $B_i(n) = 1$). Now that n is a follower for P_σ, continue running the enumerations of B_i and W_e until n enters one of these. It has to enter one of them at some stage s since we are in the case that every follower is either enumerated into B_i before being realized or is realized but does not get permission to be enumerated. If n does not enter B_i before it enters W_e (making n be realized), then we know n will never enter B_i (otherwise P_σ will enumerate n into A_i). Since the B_i are not computable, this cannot happen, so P_e^i is met. □

Since all the requirements are met, we have succeeded in building our T-minimal pair. □

With this technical result at hand we can transfer properties of minimal pairs of c.e. Turing degrees to the structure of the c.e. Q-degrees.

Corollary 27. *If* $\mathbf{c}$ *is a* T*-degree bounding no* T*-minimal pair, then* $\mathbf{c}$ *also* Q*-bounds no* Q*-minimal pair, for any representative of* $\mathbf{c}$*.*

Proof. Suppose $\mathbf{c}$ Q-bounds a Q-minimal pair $\mathbf{b}_1, \mathbf{b}_2$. By Theorem 24, there is a Turing minimal pair $\mathbf{a}_1, \mathbf{a}_2$ with $\mathbf{a}_i \leq_{sQ} \mathbf{b}_i$, hence $\mathbf{a}_i \leq_Q \mathbf{c}$ and so $\mathbf{a}_i \leq_T \mathbf{c}$ so $\mathbf{c}$ bounds a T-minimal pair. □

Corollary 28. *There are* Q*-degrees* $\{\mathbf{c}_m\}_{m\in\omega}$ *such that*

i) For every m*,* $\mathbf{c}_m$ *is non-computable.*

ii) For every m*,* $\mathbf{c}_m$ *does not bound a* Q*-minimal pair.*

iii) For every $m \neq n$*,* $\mathbf{c}_m$ *and* $\mathbf{c}_n$ *are a* Q*-minimal pair.*

Proof. Ambos-Spies and Soare showed the existence of Turing degrees with all the above-mentioned properties in the Turing degrees. We claim that the Q-degrees of these degrees have the same properties in the Q-degrees. Properties *i*) and *ii*) are immediate (the second being the above corollary), and *iii*) is also immediate because if $\mathbf{c}_m$ and $\mathbf{c}_n$ are a T-minimal pair they are also a Q-minimal pair. □

Corollary 29. *For every $n \geq 2$, there is $\mathbf{b}_n = \mathbf{c}_0 \cup \ldots \cup \mathbf{c}_{n-1}$ where $\mathbf{c}_i$ are as in the above corollary and such that $\mathbf{b}_n$ bounds n distinct nonzero degrees which pairwise form Q-minimal pairs, but $\mathbf{b}_n$ does not bound $n+1$ such degrees.*

Proof. Let $\mathbf{b}_n = \mathbf{c}_0 \cup \ldots \cup \mathbf{c}_{n-1}$ be the Turing degree as in Ambos-Spies and Soare with the above-mentioned properties in the Turing degrees. We claim that the Q-degree of $\mathbf{b}_n$ has these properties in the Q-degrees also. First, since $\mathbf{b}_n$ is the join of the $\mathbf{c}_i$, it Q-bounds all the $\mathbf{c}_i$, which by the previous corollary are pairwise Q-minimal pairs, so the first half is immediate. For the second property, we illustrate the case for $n = 2$, and this can clearly be extended for larger n with more applications of Theorem 24. Suppose towards a contradiction that $\mathbf{a}_0, \mathbf{a}_1, \mathbf{a}_2$ are Q-degrees that are below $\mathbf{b}_2$ and pairwise form Q-minimal pairs. By Theorem 24, we have degrees $\mathbf{c}_0 \leq_Q \mathbf{a}_0$ and $\mathbf{c}_1 \leq_Q \mathbf{a}_1$ which form a T-minimal pair. Note that $\mathbf{c}_0$ forms a Q-minimal pair both with $\mathbf{a}_1$ and $\mathbf{a}_2$, since if $X \leq_Q \mathbf{c}_0 \leq_Q \mathbf{a}_0$ and $X \leq_Q \mathbf{a}_1$ or $X \leq_Q \mathbf{a}_2$ then X must be computable because the $\mathbf{a}_i$ form Q-minimal pairs. Apply the theorem again to obtain $\mathbf{c}_0' \leq_Q \mathbf{c}_0 \leq_Q \mathbf{a}_0$ and $\mathbf{c}_2 \leq_Q \mathbf{a}_2$ which form a T-minimal pair. We have that $\mathbf{c}_0'$ and $\mathbf{c}_1$ are a T-minimal pair (and hence also a Q-minimal pair) because $\mathbf{c}_0$ and $\mathbf{c}_1$ are a T-minimal pair, and $\mathbf{c}_1$ and $\mathbf{c}_2$ are also a Q-minimal pair. Apply the theorem another time to obtain $\mathbf{c}_1' \leq_Q \mathbf{c}_1$ and $\mathbf{c}_2' \leq_Q \mathbf{c}_2$ which are a T-minimal pair. Now we have $\mathbf{c}_0', \mathbf{c}_1', \mathbf{c}_2'$ which pairwise form *Turing* minimal pairs, and are all Turing below the original $\mathbf{b}_2$, contradicting that $\mathbf{b}_2$ does not bound three degrees that are pairwise T-minimal pairs. For larger n, the same method with more applications of the theorem can be used to obtain the same result. Thus, $\mathbf{b}_n$ cannot bound $n+1$ degrees that pairwise form Q-minimal pairs. □

Corollary 30. *The c.e. Q-degrees have infinitely many 1-types.*

Proof. For all $m, n \in \omega$, $m > n \geq 2$, the 1-types of $\mathbf{b}_n$ and $\mathbf{b}_m$ from the above corollary are distinct. □

References

[1] Klaus Ambos-Spies. Contiguous r.e. degrees. In *Computation and proof theory (Aachen, 1983)*, volume 1104 of *Lecture Notes in Math.*, pages 1–37. Springer, Berlin, 1984.

[2] Klaus Ambos-Spies, Carl G. Jockusch, Jr., Richard A. Shore, and Robert I. Soare. An algebraic decomposition of the recursively enumerable degrees and the coincidence of several degree classes with the promptly simple degrees. *Trans. Amer. Math. Soc.*, 281(1):109–128, 1984.

[3] Klaus Ambos-Spies and Robert I. Soare. The recursively enumerable degrees have infinitely many one-types. *Ann. Pure Appl. Logic*, 44(1-2):1–23, 1989.

[4] I. I. Batyrshin. Noncancellable, singular and conjugate degrees. *Algebra Logika*, 56(3):275–299, 2017.

[5] O. V. Belegradek. Algebraically closed groups. *Algebra i Logika*, 13:239–255, 363, 1974.

[6] Sapir Ben-Shahar, Rodney G. Downey, and Mariya Soskova. On quasi-reducibility for c.e. sets Part I. The structure of the Q-degrees and the sq-degrees. *J. Logic Comput.*, 36(1):Paper No. exaf076, 27, 2026.

[7] P. Cholak, R. Downey, and M. Stob. Automorphisms of the lattice of recursively enumerable sets: promptly simple sets. *Trans. Amer. Math. Soc.*, 332(2):555–570, 1992.

[8] A. N. Dëgtev. tt- and m-degrees. *Algebra i Logika*, 12:143–161, 243, 1973.

[9] R. Downey, G. LaForte, and A. Nies. Computably enumerable sets and quasi-reducibility. *Ann. Pure Appl. Logic*, 95(1-3):1–35, 1998.

[10] R. Downey and R. A. Shore. Degree theoretic definitions of the low_2 recursively enumerable sets. *The Journal of Symbolic Logic*, 60:727–756, 1995.

[11] R. G. Downey. Recursively enumerable m- and tt-degrees. II. The distribution of singular degrees. *Arch. Math. Logic*, 27(2):135–147, 1988.

[12] R. G. Downey and C. G. Jockusch, Jr. T-degrees, jump classes, and strong reducibilities. *Trans. Amer. Math. Soc.*, 301(1):103–136, 1987.

[13] R.G. Downey and M. Stob. Structural interactions of the recursively enumerable t- and w-degrees. *Annals of Pure and Applied Logic*, 31:205–236, 1986.

[14] Rodney G. Downey and Steffen Lempp. Contiguity and distributivity in the enumerable Turing degrees. *J. Symbolic Logic*, 62(4):1215–1240, 1997.

[15] J. T. Gill and P. H. Morris. On subcreative sets and S-reducibility. *J. Symbolic Logic*, 39:669–677, 1974.

[16] C. G. Jockusch, Jr. Semirecursive sets and positive reducibility. *Trans. Amer. Math. Soc.*, 131:420–436, 1968.

[17] Carl Groos Jockusch, Jr. *REDUCIBILITIES IN RECURSIVE FUNCTION THEORY*. ProQuest LLC, Ann Arbor, MI, 1966. Thesis (Ph.D.)–Massachusetts Institute of Technology.

[18] Richard E. Ladner and Leonard P. Sasso, Jr. The weak truth table degrees of recursively enumerable sets. *Ann. Math. Logic*, 8(4):429–448, 1975.

[19] S. S. Marčenkov. A certain class of incomplete sets. *Mat. Zametki*, 20(4):473–478, 1976.

[20] J. Myhill. Creative sets. *Z. Math. Logik Grundlagen Math.*, 1:97–108, 1955.

[21] P. Odifreddi. Strong reducibilities. *Bull. Amer. Math. Soc. (N.S.)*, 4(1):37–86, 1981.

[22] R. Sh. Omanadze. Q-reducibility and nowhere nonsimple sets. *Soobshch. Akad. Nauk Gruzin. SSR*, 127(1):29–32, 1987.

[23] R. Sh. Omanadze. On the upper semilattice of recursively enumerable sQ-degrees. *Algebra i Logika*, 30(4):405–413, 507, 1991.

[24] R. Overbeek. The representation of many-one degrees by the word problem for Thue systems. *Proc. London Math. Soc. (3)*, 26:184–192, 1973.

[25] E. L. Post. Recursively enumerable sets of positive integers and their decision problems. *Bull. Amer. Math. Soc.*, 50:284–316, 1944.

[26] Richard A. Shore. Nowhere simple sets and the lattice of recursively enumerable sets. *J. Symbolic Logic*, 43(2):322–330, 1978.

[27] Robert I. Soare. Computational complexity, speedable and levelable sets. *J. Symbolic Logic*, 42(4):545–563, 1977.

[28] V. D. Solov'ev. Q-reducibility, and hyperhypersimple sets. In *Probabilistic methods and cybernetics, No. X-XI (Russian)*, pages 121–128. Izdat. Kazan. Univ., Kazan, 1974.

[29] A. M. Turing. On Computable Numbers, with an Application to the Entscheidungsproblem. *Proc. London Math. Soc. (2)*, 42(3):230–265, 1936.

[30] A. M. Turing. Systems of Logic Based on Ordinals. *Proc. London Math. Soc. (2)*, 45(3):161–228, 1939.